Titre: Title:	Fabrication d'horaires personnalisés maximisant le nombre d'heures travaillées par ordre de seniorité
Auteur: Author:	Lucas Bancel
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Bancel, L. (2017). Fabrication d'horaires personnalisés maximisant le nombre d'heures travaillées par ordre de seniorité [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2769/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2769/
Directeurs de recherche: Advisors:	Guy Desaulniers
Programme: Program:	Maîtrise recherche en génie industriel

UNIVERSITÉ DE MONTRÉAL

FABRICATION D'HORAIRES PERSONNALISÉS MAXIMISANT LE NOMBRE D'HEURES TRAVAILLÉES PAR ORDRE DE SENIORITÉ

LUCAS BANCEL
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL)
SEPTEMBRE 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

FABRICATION D'HORAIRES PERSONNALISÉS MAXIMISANT LE NOMBRE D'HEURES TRAVAILLÉES PAR ORDRE DE SENIORITÉ

présenté par : BANCEL Lucas
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. HERTZ Alain, Doctorat ès Sc., président
M. DESAULNIERS Guy, Ph. D., membre et directeur de recherche
M. AUDET Charles, Ph. D., membre

REMERCIEMENTS

En premier lieu, je tiens à remercier mon directeur de recherche : Guy Desaulniers. C'est lui qui m'a permis de travailler sur ce projet. Il m'a à la fois financé et aidé tout au long de cette épreuve. Sans lui, je ne serais certainement pas là où je suis actuellement. Ce fut un grand plaisir de travailler en collaboration avec Guy, et un profond enrichissement.

Merci à François Lessard pour sa patience et sa dévotion pendant l'ensemble du projet. Il a résolu avec moi de manière rigoureuse l'ensemble des problèmes qui se sont présentés sur notre chemin et ce fut un réel plaisir de travailler avec lui. Je lui dois beaucoup dans l'accomplissement de cette maîtrise. Merci également à Chérifa Saâdi et Rachid Hassani d'avoir pris le temps de me parler de leurs travaux et de m'avoir inculqué leur savoir.

Bien sûr, je tiens à remercier Alain Hertz pour présider mon jury de maîtrise. Merci également à M. Charles Audet d'avoir accepté d'être membre de ce même jury.

Je souhaite également remercier toutes les personnes du GERAD qui ont partagé mon quotidien pendant ces deux années passées à leurs côtés pour les moments de détente bienvenus qu'ils m'ont apportés. Je remercie notamment Lucie Desfontaines, Paul Javal et Arnaud Augustin. Je souhaite remercier mes amis de Montréal et en particulier mes colocataires Marine Bouguereau, Pierre Escoffres et Chloé Hérard, ainsi que la communauté Gadzart de Montréal. Je remercie évidemment l'ensemble de ma famille pour m'avoir accompagné dans ce projet et avoir participé de manière significative à la réussite de mes études depuis 10 ans.

RÉSUMÉ

En règle générale, une entreprise cherche à minimiser les coûts de gestion du personnel en respectant les normes et les conventions collectives. Néanmoins, il arrive que l'on décide d'ajouter une composante non négligeable aux coûts, à savoir les préférences des employés. Ainsi, on modifie légèrement les objectifs d'optimisation, en donnant de l'importance aux volontés des employés, en donnant la priorité aux employés dont l'ancienneté est la plus importante. En effet, on peut se rendre compte qu'il y a plus de bénéfices à tirer d'employés dont les préférences sont satisfaites, plutôt que d'employés choisis de façon à optimiser les coûts, mais qui sont mécontents de leurs affectations.

Ce projet de maîtrise a pour but de répondre à ce problème de fabrication d'horaires de personnel avec priorités. Nous appliquons ce problème à des magasins de vente au détail. L'objectif final est de proposer une méthode heuristique basée sur un modèle de programmation en nombres entiers permettant d'obtenir une solution optimale au problème. Nous résolvons le problème de façon séquentielle en utilisant des groupes d'employés. Par exemple nous prenons les 5 employés les plus seniors et nous optimisons l'horaire. On fixe leur horaire et on passe au groupe suivant. Il faut faire attention à ce que les algorithmes développés pour produire les solutions ne nécessitent pas des temps de calcul trop longs à l'exécution.

Dans le cadre de cette étude, une seule préférence est prise en compte. Nous tenons ainsi à ce que les employés travaillent le plus d'heures possible. Par conséquent, nous vérifierons que les employés les plus seniors travaillent plus que leurs homologues plus juniors. Un jeu de données factice permet de constater l'efficacité de notre méthode. Mais dans les instances de la vie réelle, les disponibilités rendent impossible la satisfaction des préférences au sens strict. Nous entendons par là qu'il est impossible pour certains employés seniors de travailler plus que des juniors.

D'où le besoin d'établir un évaluateur qui certifie que nous satisfaisons bien les préférences. Nous testons ensuite notre approche sur différents jeux de données et estimons l'impact de la satisfaction des préférences sur les fonctions de coût déjà existantes. Puis deux autres approches sont testées pour les comparer à celle d'origine. La première évite l'utilisation des quarts anonymes en relaxant l'intégrité sur les variables associées aux employés plus juniors. La seconde ajoute une contrainte au problème indiquant le nombre d'heures minimum que les employés seniors doivent faire. Enfin, des stratégies de réduction des temps de calcul sont proposées ainsi qu'une analyse de sensibilité pour la meilleure approche.

ABSTRACT

Traditionally, firms aim to minimize the costs linked to the management of staff while still respecting employment law and collective agreement. However, another factor, the satisfaction of preferences, can also be added to these costs. In so doing optimization is modified by taking into account senior employees wishes. Indeed employees who are happy with their schedules generally benefit the company.

This thesis will deal with the problem of staff scheduling which considers preferences and priorities and will adress real-life instances from the retail industry. The final goal is to propose a heuristic method based on integer programming in order to obtain optimal solutions. The problem will be solved sequentially using groups of employees. For instance, we will begin by building the schedule of the five most senior employees and then we will do the same for the next group. The computational time will be restricted to be between 0 and 10 minutes maximum.

In this study we will only take into account one preference: employees want to work a maximum of hours in a week. Consequently we will check if senior employees work more than junior employees. An artificial instance will be used to test the efficiency of the proposed method. In real life, employee availability cannot ensure the strict satisfaction of preferences – in other words it is possible that some senior employees work less than juniors.

So we need to create a tool that guarantees the satisfaction of preferences. We test the proposed solution approach on different instances and then estimate the impact of preferences on the cost of solutions. We will also test two other approaches and compare them to the first one. The first alternative approach avoids the use of open shifts by relaxing the integrality requirements on the variables associated with the juniors. The second one adds a constraint to the problem, indicating the number of hours that the seniors need to work, which is calculated by the first solution. Finally we propose strategies to reduce the computational times and an analysis of sensitivity for the best approach.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

PBS	Preferential Bidding System
MIP	Mixed Integer Program
MACBETH	Measuring Attractiveness by a Categorical Based Evaluation Technique
AMACSN	Affine Multi-Attribute Correlated Social Normalization
$\mathcal{H}$	ensemble des dates
$\mathcal{E}$	ensemble des employés
$\mathcal{W}$	ensemble des activités
$\mathcal{S}_e^h$	ensemble des quarts d'un employé
$\mathcal{S}^A$	ensemble des quarts anonymes
$\mathcal{K}^V$	ensemble des paliers de la fonction croissante de sur-couverture
$\mathcal{K}^U$	ensemble des paliers de la fonction croissante de sous-couverture
b_s	période de début d'un quart s
l_s	longueur en périodes d'un quart s
n_k^V	nombre de sur-couvertures sur chaque palier k
c_k^V	coût par sur-couverture dans chaque palier k
n_k^U	nombre de quarts anonymes sur chaque palier k
c_k^U	coût par quart anonyme dans chaque palier k
c_s^Y	coût fixe par quart anonyme dans la solution
c	salaire unique pour chaque employé
n^O	nombre de jours de repos minimum sur l'horizon
n^R	durée en périodes du repos minimum entre deux quarts consécutifs
n^P	nombre de périodes de travail maximum par semaine
λ_e	coefficient de seniorité pour chaque employé
γ_{es}^h	nombre d'heures travaillées par l'employé e durant le quart s à la date h
$m(x)$	nombre maximum de quarts d'une durée d'au plus x périodes
nb_{jours}	nombre de jours autorisés à travailler dans la semaine
d_{max}	durée maximum en périodes d'un quart de travail
TaQ	Temps de calcul avec retrait de Quarts
TaR	Temps de calcul avec Relaxation de la contrainte de repos minimum
TQR	Temps de calcul avec retrait de Quarts et Relaxation de la contrainte de repos minimum
v	paramètre limite des coûts réduits des quarts

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

La fabrication d'horaires constitue une problématique abordée dans de très nombreux domaines : que ce soit dans les transports aériens ou terrestres, dans le milieu de la santé ou dans la vente au détail, les compagnies ont besoin d'outils pour gérer leurs effectifs. L'objectif est alors de pouvoir générer des horaires réalisables pour tous les employés. Cela se fait sous des contraintes importantes telles que le respect de la demande ou des conventions collectives et législations. Ainsi, il faut toujours s'assurer que le nombre d'employés est suffisant pour satisfaire la demande sans pour autant être dans l'excès. Car en effet, la masse salariale d'une entreprise est une partie majeure des coûts. Par conséquent, il faut optimiser les horaires de manière à minimiser les coûts pour l'entreprise.

Dès lors qu'une entreprise possède plus d'une dizaine d'employés, il n'est plus possible de faire les horaires dans des temps raisonnables et de manière optimale à la main. Ainsi, il faut mettre en place un logiciel d'optimisation dédié. Par la suite, nous ne nous intéresserons qu'au cas de la vente au détail.

1.1 Problème étudié et contexte de l'étude

Ce projet s'inscrit dans le cadre d'un partenariat avec une entreprise qui fournit les jeux de données, une partie des codes sources utilisés, et le financement du projet. Pour cette société, nous avons pour but d'étudier comment planifier des horaires en tenant compte des préférences des employés. Donc, nous ne chercherons pas seulement à minimiser les coûts, mais aussi à satisfaire au mieux les préférences des employés. En effet, on remarque dans les milieux industriels, que plus un employé est satisfait, plus il est enclin à être productif. Pour répondre au problème, il faut également satisfaire de nombreuses contraintes qui rendent la résolution très difficile.

Premièrement, il faut satisfaire la demande. Chaque entreprise prévoit en avance la demande à satisfaire, de façon empirique. Celle-ci se compte en nombre d'employés nécessaires pour s'occuper de la clientèle. Ainsi, un nombre trop faible d'employés face à la demande entraîne un mécontentement et une perte potentielle des clients. À l'inverse, trop d'employés pour une clientèle restreinte est contre productif et coûte cher pour rien. Un exemple de solution avec trop ou pas assez d'employés est illustré figure 1.1. Dans cet exemple, il n'y a que 8 employés qui travaillent donc pour la période où la demande est égale à 9, on a de la sous-couverture (déficit d'employé, en rouge). Pour la période 6, on se rend compte qu'il y a un employé en trop qui travaille, il y a donc de la sur-couverture (orange).

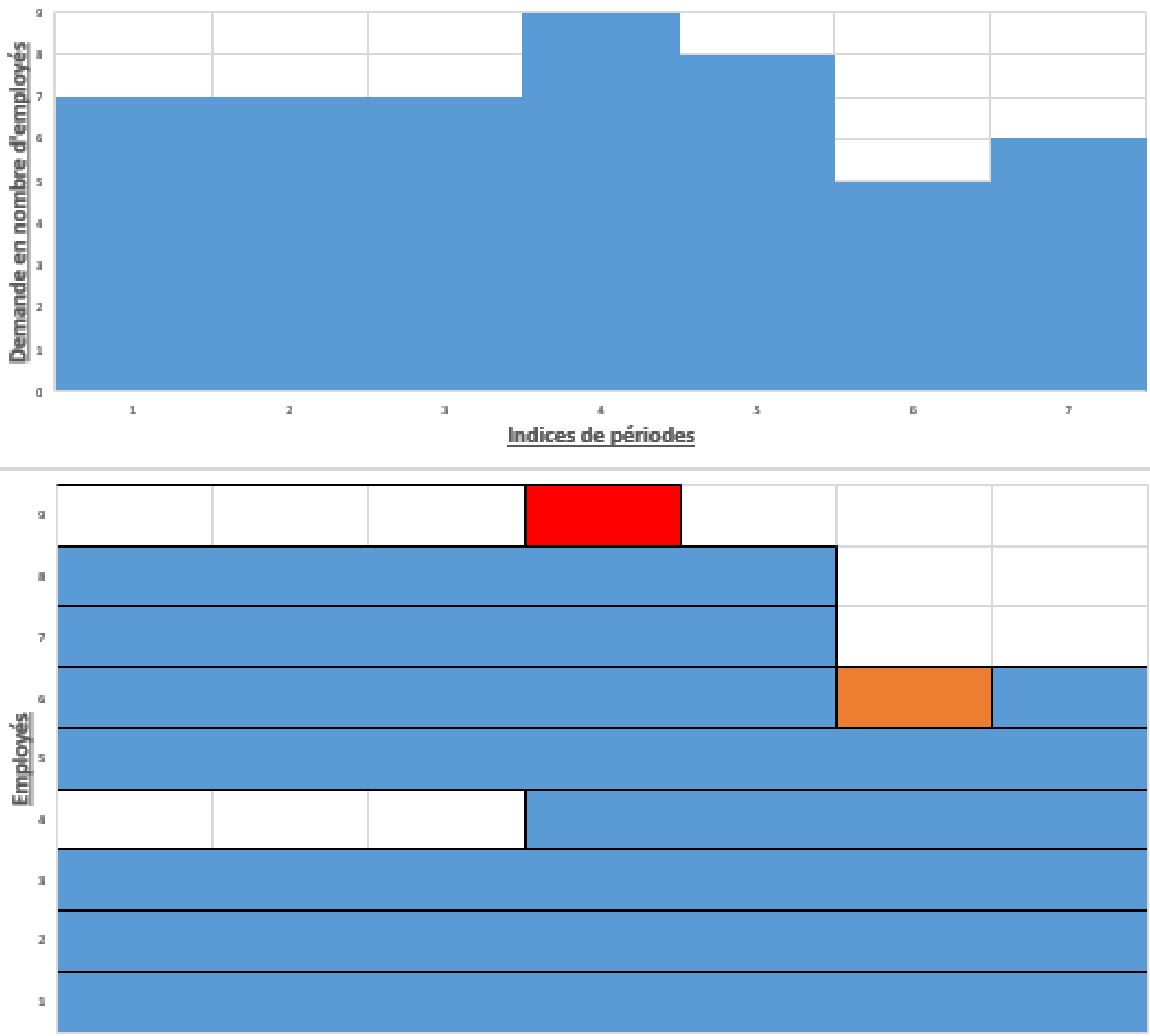

Figure 1.1 Exemple de solution avec sur-couverture et sous-couverture

Deuxièmement, il faut faire attention à ce qu'un employé fasse le travail pour lequel il est qualifié. Par exemple, nous ne donnerons pas un travail de caissier à un employé n'étant capable de faire que de la mise en rayon. Nous devons nous assurer d'affecter les activités aux employés adéquats.

Enfin, il faut prendre en compte les lois qui régissent le travail. On se doit alors de respecter le temps de repos entre deux jours de travail, comme le nombre de jours de congés minimum dans une semaine ou la durée totale de travail dans une semaine.

Pour notre étude, l'horizon de planification sera d'une semaine. Nous ne considérons pas le temps comme continu, mais comme discret. En effet, nous découpons l'horizon de planification en périodes. Nous supposerons la demande connue en tout temps, et nous ne considérerons pas de perturbations possibles, par exemple le retard d'un employé. D'ordinaire, les préférences des employés sont utilisées comme une contrainte supplémentaire. Pour nous, ce sera une composante de la fonction objectif. De plus, nous nous attacherons à maximiser les préférences en donnant la priorité aux employés dont l'ancienneté est la plus importante. Par ancienneté, ou seniorité, on entend que l'employé le plus senior est celui qui est dans la société depuis le plus longtemps. C'est-à-dire que nous ne chercherons pas à produire des horaires équitables. Nous nous intéressons au cas d'un seul département comportant entre 25 et 50 employés avec différents métiers. Les pauses ne sont pas prises en compte.

1.2 Motivation et démarche

Il existe divers logiciels d'optimisation d'horaires dans le milieu de la vente au détail. La plupart du temps, ils n'intègrent pas la composante préférence des employés à l'optimisation et se contentent de minimiser les coûts. Les autres tiennent compte en général de la préférence des employés comme une contrainte et non comme un objectif. Par exemple, ils imposent que chaque employé ait une satisfaction minimum à l'issue de l'optimisation. Cette satisfaction minimum peut être un pourcentage qui est variable d'un employé à l'autre. Nous voulons non pas tenir compte de la préférence comme d'une contrainte, mais bien comme d'un objectif. C'est-à-dire que nous ne cherchons plus seulement à minimiser les coûts, mais bien à minimiser les coûts tout en essayant de satisfaire au mieux les préférences. Ainsi, peu importe que certains employés ne soient pas du tout satisfait, tant que la satisfaction globale a été maximisée. Ce projet se démarque en mettant la priorité des employés les plus anciens au cœur de l'optimisation. Le but est donc d'intégrer cette nouvelle composante aux logiciels actuels. Ceci résulte de la volonté des entreprises d'améliorer la productivité des employés, dans le but de générer plus de bénéfices à long terme.

Pour réaliser le projet, nous procédons en plusieurs étapes. En premier lieu, nous nous attelons à mettre en place un modèle prenant en compte une seule préférence : le nombre d'heures travaillées. Grossièrement, plus un employé est ancien, plus il veut travailler un nombre d'heures conséquent. Dès cette étape, nous pouvons être confrontés à des problèmes liés aux disponibilités et qualifications des employés. En effet, si l'employé le plus senior n'a que 4 heures de disponibilité dans la semaine, et que nous lui attribuons ces 4 heures, nous ne voulons a priori pas qu'un employé moins senior travaille plus. Or, pour respecter la contrainte liée à la demande, il est nécessaire d'attribuer un nombre d'heures plus important à des employés plus juniors. Nous allons donc évaluer nos solutions pour affirmer que chaque employé a été satisfait au maximum. Pour ce faire, nous allons établir un ratio entre le nombre d'heures travaillées par les employés et le nombre maximum d'heures qu'ils auraient pu travailler. Ainsi, un employé qui a un ratio de 1 sera satisfait à 100%, autrement dit, nous n'aurions pas pu lui attribuer plus d'heures. Cela assurera que la seniorité est respectée finalement.

Par la suite, l'objectif est de tester de nouvelles approches de résolution ainsi que des stratégies de réduction des temps de calcul. Plus précisément, comme nos approches de résolutions sont heuristiques, il est fort possible qu'il y ait des imperfections lors de la résolution. Par exemple, un employé relativement senior peut ne pas être satisfait à 100%. Cela est rendu possible par le caractère séquentiel de notre approche, qui la rend heuristique. Dans ce cas, il est intéressant de trouver des leviers d'actions pour modifier notre approche initiale et comparer ces approches. Aussi, le temps de calcul étant un aspect très important, il est intéressant de trouver des stratégies pour le réduire.

1.3 Organisation du mémoire

Dans le prochain chapitre, nous effectuons un état de l'art permettant de retracer les diverses méthodes utilisées jusqu'à maintenant. Celle-ci se déroule en deux temps, le premier consacré à la génération d'horaires du personnel, et le second au traitement des préférences dans la littérature.

Le chapitre 3 sert à présenter le problème de manière formelle ainsi que les modèles utilisés et la première approche de résolution. Dans le chapitre 4, nous passons en revue les diverses méthodes et approches que nous avons implémentées pour améliorer le travail réalisé dans le chapitre 3. On compte deux nouvelles approches, ainsi que diverses stratégies de réduction des temps de calcul. Le chapitre 5 offre une analyse de sensibilité. Cette dernière porte sur les paramètres que l'on fait varier pour la meilleure approche, tandis que la conclusion constitue le chapitre 6.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre a pour but de présenter un état de l'art des travaux déjà effectués sur le problème décrit précédemment. Nous commençons par développer les recherches effectuées sur la fabrication d'horaires de personnel en général (section 2.1), puis nous nous intéressons plus particulièrement aux préférences des employés dans la littérature (section 2.2).

2.1 Génération d'horaires de personnel

Cela fait plus d'un demi-siècle que les problèmes de fabrication d'horaires sont traités en recherche opérationnelle. C'est Dantzig qui y répondit le premier en 1954 (Dantzig (1954)). Il répondait à un article d'Edie (1954) qui avait entrepris d'optimiser le fonctionnement des péages de New York. Particulièrement, il s'est intéressé au temps d'attente des véhicules qui s'y présentaient. Edie proposa alors une heuristique pour trouver une solution au problème. En réponse, Dantzig opta pour un programme linéaire en nombres entiers. Celui-ci est basé sur un problème classique : le problème de recouvrement. Il est alors assez aisé de résoudre le problème soumis par Edie, dans la mesure où la plupart des contraintes sont intégrées en pré-optimisation dans l'énumération des différents quarts de travail réalisables. Cette façon de procéder fonctionne très bien sur des cas simples, mais demeure beaucoup plus compliquée dès lors que l'on augmente le nombre de variables : en effet, considérer les pauses ou bien plusieurs activités à gérer en même temps a tendance à augmenter énormément la taille du modèle initial. Par conséquent, il devient assez difficile à résoudre, bien que sa formulation soit assez simple. Après cet échange, plusieurs chercheurs se sont intéressés au problème, dans le but d'y apporter de la flexibilité. Parmi ces flexibilités, on note surtout le cas multi-activités et l'ajout des pauses. Par exemple, Bechtold et Jacobs (1990) ont proposé un modèle implicite permettant l'ajout d'une pause de durée fixée à l'intérieur du quart de travail, en considérant un environnement mono-activité.

Plus tard, Aykin (1996) a généralisé le modèle établi par Dantzig. Il s'est intéressé au cas de pauses multiples dans des fenêtres de temps. Il utilise des variables pour chaque fenêtre où une pause peut commencer. En l'état, cette approche a besoin de beaucoup moins de variables que l'approche de Dantzig (1954). Néanmoins, cette approche concerne plus aisément le contexte mono-activité, car dans le cas multi-activités, l'énumération des quarts devient difficile.

Ernst et al. (2004) et Van den Bergh et al. (2013) proposent des états de l'art complets. On choisit celui d'Ernst et al. pour présenter les différents travaux effectués sur le sujet. Il divise le problème en six étapes : la modélisation de la demande, les journées de congé, la construction et sélection des quarts de travail, la construction de lignes de travail, l'affectation des tâches et l'affectation des employés. Cette division est due au fait qu'il est très difficile de traiter ces six problèmes à la fois, surtout considérant les capacités de calcul des ordinateurs actuels. Néanmoins, selon le contexte de l'entreprise, il est possible de combiner plusieurs de ces modules, voire d'en occulter au besoin. Dans le cadre de la vente au détail, pour fabriquer un horaire, on s'emploie à :

— modéliser la demande pour chaque activité ;
— déterminer les jours de travail et de repos de chaque employé ;
— créer et affecter les quarts de travail ;
— affecter les tâches aux quarts de travail.

Le milieu de la vente au détail est un environnement multi-activités, c'est-à-dire que les employés peuvent travailler sur plusieurs postes différents selon leurs qualifications. Cela s'oppose à l'environnement mono-activité pour lequel tous les employés font peu ou prou la même activité. C'est dans l'environnement multi-activités que Lequy et al. (2012) distinguent deux catégories pour le travail des employés. D'un côté, ils définissent le travail que l'on peut interrompre comme tenir une caisse de supermarché. De l'autre, ils définissent le travail qu'on ne peut interrompre comme assister une opération pour une infirmière. Dans notre cas, on s'intéressera majoritairement au travail qui peut être interrompu, car il correspond à ce que l'on retrouve dans la vente au détail.

Omari (2003) et Vatri (2003) se sont intéressés à ce type de tâche pour les contrôleurs aériens. Omari utilise une méthode heuristique pour affecter les activités aux contrôleurs aériens. Celle-ci repose sur un modèle de réseau mutli-flots avec des variables de ressources. Dans son cas, une unité de flot correspond à un quart de travail. Il découpe l'horizon de planification en intervalles de temps et résout chaque problème (un par intervalle) par une décomposition de Dantzig-Wolfe. Vatri étend le modèle d'Omari. Il permet la construction des quarts et l'affectation des activités simultanément. Il résout également le problème à l'aide d'une heuristique basée sur de l'énumération implicite et de la génération de colonnes.

Demassey et al. (2005) ont aussi travaillé sur le cas multi-activités. Dans leurs travaux, ils ne scellent pas les quarts en amont et doivent régir les pauses. Ils utilisent une approche de génération de colonnes pour la construction des horaires réalisables en introduisant une contrainte forçant l'outil d'optimisation à ne générer que des horaires de coût réduit non positif. Cette contrainte est appelée *cost-regular global constraint*. La programmation par contraintes a été largement utilisée pour diminuer la taille des programmes mixtes en nombres entiers liés à la construction de quarts de travail anonymes. C'est ainsi que Côté et al. (2007) intègrent la contrainte *cost-regular global constraint* dans des programmes mixtes. En comparant leurs résultats avec d'autres obtenus sans utiliser cette nouvelle contrainte, ils constatent que les leurs sont bons.

Puis Bouchard (2008) résout le problème d'Omari, cette fois de manière exacte. Il utilise la coloration de graphe pour résoudre ce problème de flots. Il parvient à obtenir une solution optimale, dans un temps de calcul rapide, en utilisant le concept de blocs d'activités. Un bloc représente l'affectation possible d'une activité à un quart de travail.

2.2 Horaires personnalisés avec équité ou priorités

La question de la construction d'horaires avec priorités ou équité (*rostering*) s'est beaucoup posée dans le domaine aérien. Ainsi, la plupart de la littérature existante traite d'applications à la fabrication d'horaires mensuels pour les équipages d'avion. L'objectif général est soit de maximiser les préférences des employés dont l'ancienneté est la plus importante, soit d'introduire la notion d'équité entre les employés. Cet aspect génère des problèmes très difficiles à résoudre, et ce n'est pas celui auquel on s'intéresse dans le présent mémoire. Néanmoins, un rapide état de l'art de ce mode de construction d'horaires permet d'établir les idées communes entre la construction avec équité et celle avec priorités.

2.2.1 Horaires personnalisés avec équité

Pour ce mode de fabrication d'horaires, on attribue des poids aux activités souhaitées, dans le but d'essayer de forcer leur attribution à des employés qui en ont fait la demande. Finalement, on cherche à maximiser la satisfaction moyenne des employés. La difficulté de ce problème pousse à envisager une vision à long terme de la notion d'équité, c'est-à-dire qu'on ne s'intéresse pas au fait d'obtenir une équité parfaite pour un horizon de planification. Mais on cherche à obtenir l'équité sur plusieurs périodes de planification.

Pour résoudre ce problème, Glanert (1984) propose un modèle d'affectation. Le but est de hiérarchiser chaque tâche et chaque employé selon leur priorité. Puis, il faut attribuer la tâche de priorité maximum à l'employé de priorité maximum. Une autre façon de faire, est celle de

Ryan (1992), qui utilise comme Dantzig, un problème de recouvrement. La génération des horaires réalisables se fait en amont de façon heuristique. L'équité est intégrée à la fonction objectif du problème. Cet objectif a pour but de refléter les intérêts de la compagnie, mais aussi des équipages. D'un côté, on minimise le nombre d'agents de bord, et de l'autre on essaye de satisfaire au mieux le nombre de jours de repos entre deux vols du même type.

Beaucoup d'autres chercheurs ont utilisé des méthodes heuristiques pour traiter le problème du *rostering*. C'est le cas notamment de Day et Ryan (1997). Ils utilisent un modèle de recouvrement pour générer des horaires équitables. Ils résolvent deux problèmes simultanément par génération de colonnes : l'affectation des jours de repos et des rotations.

Il ne s'agit là que de solutions heuristiques, ce qui implique qu'elles ne sont pas optimales. Pire, elles ne peuvent pas assurer que l'on se rapproche de façon acceptable de la solution optimale dès lors que les instances deviennent grandes.

Gamache et al. (1999) sont les seuls à proposer une heuristique convenable pour des problèmes de très grande taille. Pour ce faire, ils partitionnent le problème en groupe d'employés semblables. Eux également utilisent la génération de colonnes pour fabriquer les horaires des employés d'un même groupe. Cette division du problème en plusieurs groupes leur a permis de traiter des problèmes contenant jusqu'à 1000 employés dans des temps de calcul tout à fait raisonnables. L'équité est gérée dans le sous problème, celui-ci étant modélisé par un graphe. Un chemin dans celui-ci satisfait les contraintes du *rostering*. Dans le cas où la structure du réseau ne garantie pas la satisfaction de ces contraintes„ ils ajoutent des restrictions dans les problèmes de plus court chemin. Ces restrictions sont modélisées par l'ajout de variables de ressource qui grandissent tout au long du chemin.

Boubaker et al. (2010) se sont attaqués au problème en proposant un modèle de partitionnement, ainsi que deux heuristiques pour le résoudre. La première est basée sur un *branch-and-price* classique, tandis que la seconde est obtenue en combinant la première avec une méthode d'agrégation de contraintes dynamique proposée par Elhallaoui et al. (2010). La gestion de l'équité se fait cette fois-ci en amont. En effet, ils construisent des horaires anonymes qu'ils redistribuent aux employés par la suite. Ces horaires sont construits de sorte à ce que'ils contiennent à peu près le même nombre de jours de congés et le même nombre de jours de travail. Les résultats informatiques montrent que l'on améliore la qualité de la solution tout en diminuant le temps de calcul par rapport à ce qui était proposé dans le passé.

2.2.2 Horaires personnalisés avec priorités

Le mode de fabrication d'horaires avec priorités se différencie du *rostering* au niveau de la fonction objectif. Dans le cas du *rostering*, on cherche à optimiser la somme des satisfactions des employés. En revanche, l'objectif de la construction d'horaires avec priorités tend à maximiser la satisfaction de chaque employé. Dans beaucoup de cas, le nôtre inclus, on donne la priorité aux employés dont l'ancienneté est la plus importante. On cherche donc toujours à attribuer l'emploi du temps de score maximum à un employé senior, et on veille à ne jamais détériorer ce score au profit d'un employé plus junior. Ce problème est communément appelé *Preferential Bidding System* (PBS) dans le domaine aérien.

Cette façon de construire des horaires est assez récente, ce qui explique que peu d'articles traitant ce sujet existent.

Au sein de plusieurs compagnies aériennes, on a ressenti le besoin de travailler sur des méthodes de fabrication d'horaires personnalisés avec priorités. C'est le cas de Quantas (Moore et al. (1978)), et CP Air (Byrne (1988)). Dans ces deux cas, ils utilisent une méthode heuristique qui construit séquentiellement un bon horaire pour chaque employé, en partant du plus ancien vers le moins ancien.

Cette heuristique fonctionne en trois temps. Premièrement, il faut définir les priorités des employés sur chaque activité. Deuxièmement, il faut fabriquer séquentiellement un horaire réalisable et de bon score pour chaque employé de bord selon son ancienneté. À chaque itération, on affecte l'activité la plus désirée parmi celles restantes, i.e. celles non affectées aux itérations précédentes. On répète ce procédé jusqu'à ce qu'un horaire réalisable soit obtenu. Pour les itérations suivantes, les activités attribuées durant l'itération courante sont fixées et donc ne sont pas ré-attribuables. À chaque étape, il convient de vérifier que l'on peut couvrir toute la demande avec les employés restants. Enfin, une fois tous les horaires fabriqués, on peut procéder à des échanges d'activités entre deux employés pour post-optimiser.

L'inconvénient de ces méthodes est que pour les problèmes de grande taille, on n'arrive pas à se rapprocher suffisamment de la solution optimale. Gamache et al. (1998) proposent néanmoins une méthode plus efficace que les précédentes. Cette méthode n'est pas exacte, mais elle a l'avantage d'être exacte pour chaque employé compte tenu de ce qui a été fixé, ce qui n'est pas le cas dans les autres méthodes proposées dans la littérature. Cette méthode se base sur une approche séquentielle fondée sur l'ancienneté. On construit l'horaire des employés les uns après les autres en partant du plus ancien au moins ancien. A une itération donnée, on maximise le score de l'employé courant. Ce score est évalué par la somme des poids accordés à chaque activité. Les poids peuvent varier d'un employé à l'autre en fonction des préférences qu'ils ont énoncées.

Ainsi, pour un employé donné et tous les horaires des employés le précédant donnés, on résout un programme en nombres entiers par une méthode de génération de colonnes dans un processus de *branch and bound*. Il s'agit d'affecter à cet employé des rotations parmi celles qu'il reste à attribuer. On obtient alors un horaire optimal pour cet employé, le tout en demeurant sûr qu'il existe des horaires réalisables pour les employés suivants.

Certaines recherches plus récentes ont amélioré les temps de calcul, ce qui implique que l'on peut s'intéresser à des instances de plus grandes taille. C'est notamment le cas de Jeandroz (2001) qui utilise une heuristique durant la génération de colonnes en usant de compteurs dynamiques. Ceux-ci servent d'un côté à évaluer l'offre et la demande en employés, et d'un autre à relaxer certaines contraintes. De plus, l'horizon de planification est divisé en intervalles. Chaque extrémité correspond au début et à la fin d'une activité. Puis on résout un problème de plus court chemin séquentiellement pour chaque employé par ordre d'ancienneté. Le plus court chemin trouvé donne alors l'horaire complet de l'employé courant. Le critère d'arrêt de cette méthode est la possibilité que l'itération suivante ne constitue pas un problème réalisable.

Une autre façon d'améliorer un peu la méthode proposée par Gamache et al. (1998) est proposée par St-Germain (2004). Elle ajoute une heuristique permettant de choisir le meilleur horaire pour un employé parmi plusieurs horaires de score équivalent. Ainsi, elle oblige à choisir l'horaire de l'employé courant qui a le moins d'impact sur les horaires des employés moins anciens. Par conséquent elle améliore sensiblement la qualité de la solution. Pour ce faire, elle effectue un calcul de pénalités de l'horaire courant selon les préférences des employés suivants.

Ce sont Achour et al. (2007) qui ont proposé la première méthode exacte pour résoudre le problème du PBS. Ils résolvent le problème de façon séquentielle en résolvant des programmes en nombres entiers pour chaque employé du plus senior au moins senior par une méthode de génération de colonnes. Au fur et à mesure que la résolution progresse, on fixe l'horaire des employés les plus anciens. Contrairement à la méthode heuristique de Gamache et al., les horaires ne sont pas forcément fixés en respectant l'ordre exact de seniorité. De plus, à chaque itération, il peut y avoir plusieurs horaires fixés comme aucun. Pour que la solution soit exacte, il faut d'abord s'assurer que la méthode vérifie les quatre critères suivants à chaque itération :

— construire un horaire réalisable ;
— couvrir entièrement la demande restante avec la construction des horaires suivants ;
— construire un horaire dont le score est maximum, tout en satisfaisant les deux premiers critères ;
— construire un horaire qui permette de choisir parmi les meilleurs horaires pour les employés les plus juniors.

Dans les méthodes précédentes, le choix trop rapide et arbitraire des horaires à fixer diminuait la qualité de la solution. Dans l'approche d'Achour et al., on ne choisit les horaires à fixer qu'une fois que l'on est capable de prouver que c'est le meilleur selon les quatre critères énoncés avant. Les résultats sur des cas réels montrent que l'on améliore de façon non négligeable la qualité des solutions obtenues par les méthodes heuristiques, le tout dans des temps de calcul semblables.

2.2.3 Horaires avec préférences

Dans le domaine de la santé, beaucoup de recherches ont été effectuées pour réaliser des horaires d'infirmiers avec des préférences. La différence majeure entre le domaine de la santé et celui de la vente au détail réside dans les quarts. Dans le premier, on travaille 24 heures par jour et les quarts sont d'une durée de 8h. Dans le second, on travaille rarement la nuit mais les quarts peuvent avoir des durées comprises entre 3 heures et 8 heures.

L'idée générale est de satisfaire les préférences individuelles en créant les horaires. Les préférences peuvent être diverses : nombre d'heures travaillées, préférer un quart plutôt qu'un autre, ne pas travailler un certain jour. Quelques méthodes exactes ont d'abord été essayées pour résoudre ce problème. Toutes utilisent un modèle de recouvrement, et la plupart utilise la génération de colonnes pour le résoudre.

Les approches heuristiques utilisées ont adopté une autre stratégie. D'abord on cherche une solution initiale, puis on l'améliore en utilisant des voisinages. Miller et al. (1976) travaillent sur une heuristique qui résout le problème par un algorithme de descente cyclique. Cela consiste à fixer les horaires de toutes les infirmières sauf une, et de chercher des horaires de meilleur coût pour celle-ci. Le caractère cyclique provient du fait qu'on réitère sur toutes les infirmières jusqu'à ce qu'aucune solution meilleure ne soit trouvée. La méthode globale ressemble à la nôtre dans la mesure où la fonction objectif minimise les coûts associés à la création des horaires, et des pénalités liées aux préférences des infirmiers.

Plus tard, Bard et Purnomo (2005) résolvent le problème en utilisant de la génération de colonnes. Le programme en nombre entiers est un problème de recouvrement pour lequel les colonnes correspondent à des horaires alternatifs qu'un infirmier pourrait réaliser. À chaque horaire, est associé un coût c_{ij} pénalisant l'affectation de l'horaire j à l'infirmier i. La valeur des c_{ij} est une fonction à la fois du nombre de préférences violées dans la solution, mais aussi du degré de violation de ses mêmes préférences.

CHAPITRE 3 PROBLÈME, MODÈLE ET MÉTHODE DE RÉSOLUTION DE BASE

Dans ce chapitre, nous commençons par présenter une définition détaillée du problème. Nous y expliquons toutes les considérations et contraintes importantes que nous prenons en compte. Ensuite, nous exposons les modèles. L'un sans préférence que l'on utilise pour obtenir nos solutions de référence. L'autre, avec l'ajout de la préférence choisie, qui constitue le cœur de ce travail. Cela nous permet de mieux comprendre les fondations sur lesquelles nous étudions ensuite d'autres approches de résolution. Enfin, nous montrons comment nous procédons pour nous assurer que notre approche fonctionne correctement et résout le problème, ainsi que les résultats liés à notre méthode de résolution.

3.1 Présentation du problème et des considérations prises en compte

Nous travaillons sur un horizon de planification d'une semaine $\mathcal{H} = \{1, \ldots, 7\}$, divisé en périodes $p \in \mathcal{P}$ de même durée égale à 15 minutes. Le travail des employés est divisé en plusieurs activités $w \in \mathcal{W}$, chacune présentant une courbe de demande indépendante des autres. Les activités peuvent être aussi diverses que de la mise en rayon ou tenir une caisse. Les quarts que nous utilisons sont des quarts mono-activité (i.e., un employé est affecté à la même activité pour toute la durée du quart), et les employés peuvent ne pas être qualifiés pour toutes les activités. Les courbes de demande associent à une période p de l'horizon de planification et une activité w, un besoin en nombre d'employés d_w^p. Dans les instances que nous utilisons, la demande est nulle pendant la nuit, typiquement de 23h à 6h. Généralement, un employé peut changer d'activité ou prendre une pause durant son quart de travail. Ici, nous supposons que chaque employé ne peut être affecté qu'à une seule activité durant un même quart de travail, et nous ne prenons pas en compte les temps de pause. Nous faisons ce choix car dans les magasins de vente au détail, les pauses sont souvent décidées au cours des opérations, dans les périodes creuses. Les courbes de demande sont ajustées à la hausse pour laisser la flexibilité nécessaire à l'ajout des pauses.

L'ensemble des employés est noté $\mathcal{E}$ et nous supposons qu'ils sont numérotés de 1 à $|\mathcal{E}|$, du plus senior au plus junior. Pour chaque employé $e \in \mathcal{E}$, nous avons à notre disposition un ensemble de quarts possibles $\mathcal{S}_e^h$, avec $h \in \mathcal{H}$, parmi lesquels nous choisissons un sous-ensemble pour composer son horaire. Les quarts possibles de chaque employé respectent ses qualifications, ses disponibilités et ses durées minimale et maximale de quart. De plus, un quart

$s \in \mathcal{S}_e^h$ est défini par une période de début b_s, une longueur en nombre de périodes l_s et une activité w_s à laquelle l'employé est affecté.

Le problème est sujet à de nombreuses contraintes qui restreignent l'espace des solutions possibles. Parmi elles, on compte des contraintes strictes comme le temps de repos minimum entre deux quarts consécutifs pour un employé, le nombre de jours de travail/congé minimum par semaine ainsi que le nombre d'heures maximum qu'un employé peut travailler pour satisfaire aux conventions collectives. Ces contraintes sont des conditions sine qua non à la faisabilité de l'horaire.

Il existe aussi des contraintes dites molles, c'est-à-dire que nous pouvons nous permettre de violer au cours de l'optimisation. Par exemple, les contraintes liées au respect de la demande en tout temps sont molles. Ainsi, on peut dépasser la demande à une période donnée, comme il est possible de ne pas la combler. Dans le premier cas, on parle de sur-couverture et dans l'autre de sous-couverture. Évidemment, ce n'est pas parce que cette contrainte est molle qu'il faut pour autant ne jamais la respecter. Donc, nous ferons en sorte de pénaliser fortement la violation de la contrainte de demande. Cela nous évite d'obtenir trop souvent des solutions non réalisables, car il est très improbable de pouvoir construire des horaires réalisables en respectant la demande. Pour pallier le problème de la sous-couverture, qui est inadmissible d'un point de vue opérationnel, nous introduisons un ensemble de quarts anonymes $s \in \mathcal{S}^A$. Ces quarts ne sont là que pour combler la demande aux périodes pour lesquelles les ressources humaines planifiés ne suffisent pas. L'objectif est de pouvoir attribuer ces quarts anonymes à des employés temporaires, ou bien de demander à certains employés de commencer leur quart plus tôt, ou encore finir leur quart plus tard. Ils sont définis de la même façon que les quarts de travail, mais ne sont pas soumis à des règles de longueur minimale ou maximale. Nous verrons par la suite que la sur-couverture comme l'utilisation des quarts anonymes sont très pénalisées, de façon à les utiliser avec parcimonie.

Enfin, le problème consiste à minimiser, au regard de toutes les contraintes énoncées, les divers coûts associés au problème. Il existe trois sources de coûts différentes : la sur-couverture, l'utilisation des quarts anonymes, et les salaires. Dans ce projet, le but est d'ajouter la satisfaction d'une préférence à la minimisation des coûts. Nous choisissons de satisfaire la préférence selon laquelle les employés doivent travailler le plus d'heures possible, dans la limite des règles issues des conventions collectives. De plus, nous voulons donner la priorité aux employés dont l'ancienneté est la plus importante. Pour modéliser cette préférence, nous décidons d'ajouter un terme à la fonction objectif qui s'apparentera à un bonus. Ainsi, si les employés travaillent le plus d'heures possible, cela diminue artificiellement le coût de la solution. Dans les sous-sections suivantes nous détaillons les principales contraintes du problème.

3.1.1 Respect de la demande

Le respect de la demande est un point capital de l'optimisation des horaires d'employés. En effet, une entreprise ne peut se permettre d'être en sous-effectif au moment de recevoir des clients. Si un client n'est pas servi comme il le désire ou si l'attente est trop longue, il se peut qu'il décide de ne plus revenir dans le magasin. C'est ainsi qu'il faut toujours s'assurer que la demande soit bien couverte. Les courbes de demande pour chaque activité (voir l'exemple à la figure 3.1) sont établies par analyse de l'historique. Puisque le problème étudié survient dans le domaine de la vente au détail, il est aisé de constater que la demande est plus importante en fin de semaine plutôt que les jours de semaine. De même, il y a plus de chances que les gens viennent en magasin en fin d'après-midi plutôt qu'en fin de matinée ou du temps de midi. De façon identique, on peut savoir quelle période de l'année est plus achalandée qu'une autre. Le mois de décembre sera sans doute plus chargé en demande que le mois de mars. Tout ceci est vérifié par l'expérience et permet d'obtenir des courbes de demande d'une bonne précision pour un horizon d'une semaine.

Dans le même ordre d'idées, un respect trop important de la demande est néfaste également. Certes, dans ce cas-là, la demande est toujours assurée d'être satisfaite. Néanmoins, cela peut induire un coût important et inutile pour l'entreprise. Nous pouvons également considérer que des employés qui ne travaillent pas ou peu, peuvent éprouver de la lassitude vis-à-vis de leur travail, ce qui serait synonyme de perte d'efficacité. C'est pourquoi, nous pénalisons assez chèrement la sur-couverture. Cela permet de l'utiliser avec précaution et seulement quand le gain associé compense la perte qui serait induite par de la sous-couverture ailleurs.

Il faut noter que la sous-couverture est interdite (en considérant que les quarts anonymes vont être affectés) et la sur-couverture très pénalisée. Ainsi, des coûts sont associés aux nombres de quarts anonymes utilisés pour combler un déficit de couverture à chaque période. De même, il existe des coûts pour chaque quart correspondant à de la sur-couverture à chaque période. Ces pénalités sont imposées comme des fonctions croissantes par morceaux. Cette fonction ne dépend pas de l'activité pour laquelle il y a de la sur-couverture et est définie par un ensemble de paliers $\mathcal{K}^V = \left\{1, ..., m^V\right\}$, un nombre n_k^V de sur-couverture à chaque palier $k \in \mathcal{K}^V$ et un

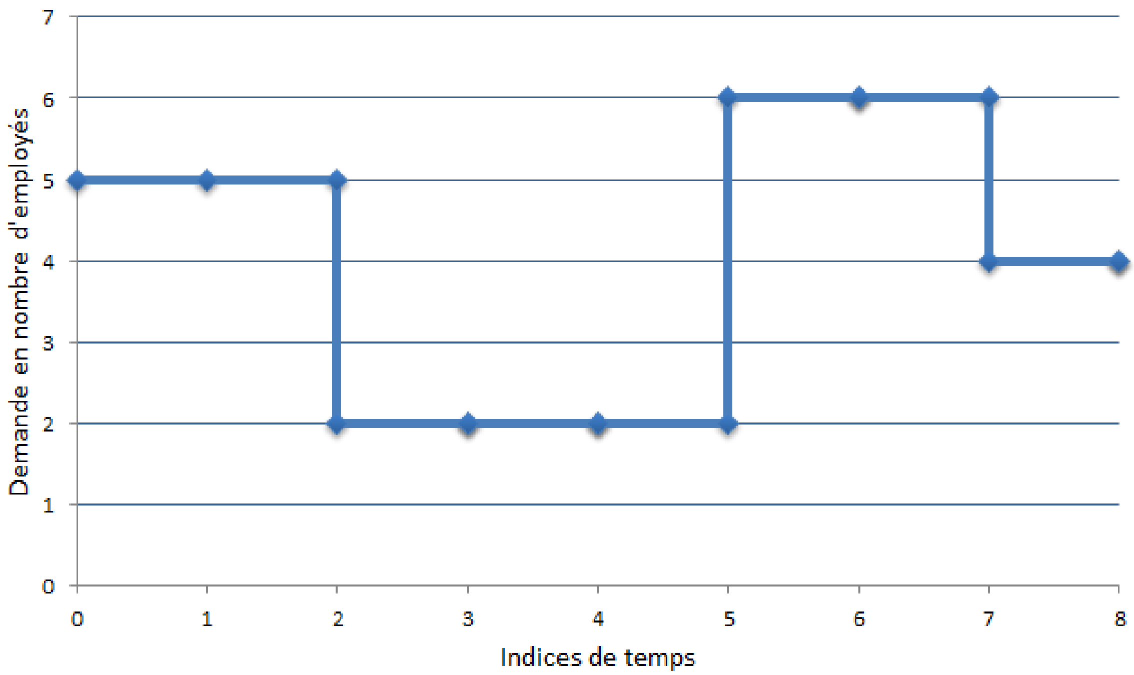

Figure 3.1 Exemple de courbe de demande

coût c_k^V par sur-couverture dans chaque palier k. Le but est alors de répartir au maximum la sous-couverture et la sur-couverture sur l'horizon plutôt que de les concentrer sur certaines périodes.

De la même façon que pour la fonction des coûts de sur-couverture, nous définissons celle des coûts de sous-couverture : $\mathcal{K}^U$ représente l'ensemble des paliers, n_k^U un nombre de quarts anonymes à chaque palier $k \in \mathcal{K}^U$ et c_k^U le coût par quart anonyme dans chaque palier k. Comme cela a été énoncé précédemment, les quarts anonymes peuvent durer plusieurs périodes. En fait, nous préférons qu'ils soient de durées similaires à celles des quarts personnalisés, car il est plus facile dans ce cas d'attribuer le quart à un employé (temporaire ou provenant d'un autre magasin). En pratique, il est difficile de donner un quart de 15 minutes ou une heure à un employé même temporaire. Par conséquent, nous attribuons un coût c_s^Y qui favorise les quarts de plus longue durée. Par exemple, au lieu d'avoir deux quarts anonymes de durée 1 heure sur deux périodes consécutives, nous préférerons avoir un seul quart d'une durée de 2 heures.

Les tableaux 3.1 et 3.2 précisent les paliers des fonctions de coûts d'utilisation des quarts anonymes et de sur-couverture.

Tableau 3.1 Coûts des quarts anonymes

Quart anonyme utilisé	Coût par minute
Premier	100
Deuxième	238
Troisième	483
Quatrième	999
Cinquième	1654
Tous à partir du sixième	2000

Tableau 3.2 Coûts des quarts en sur-couverture

Quart en sur-couverture	Coût par minute
Premier	90
Deuxième	125
Troisième	166
Quatrième	243
Cinquième	328
Tous à partir du sixième	400

3.1.2 Repos minimal entre deux quarts

Une contrainte délicate à respecter est le repos minimal entre deux quarts affectés à un même employé. Il y a traditionnellement deux manières de la modéliser. La première consiste, pour chaque paire de jours consécutifs travaillés, à indiquer que la différence entre l'heure de début du quart au deuxième jour et l'heure de fin du quart au premier jour doit être plus grande ou égale au temps de repos minimum. Cela peut cependant poser problème au niveau de la relaxation linéaire. Illustrons-le à l'aide de la figure 3.2, de façon similaire à Michon-Lacaze (2015).

Sur cette figure, nous observons trois quarts différents admissibles pour un employé donné. Si nous considérons que le repos minimal doit être de 12h, les quarts a et b sont incompatibles, tandis que les quarts a et c peuvent être choisis tous deux.

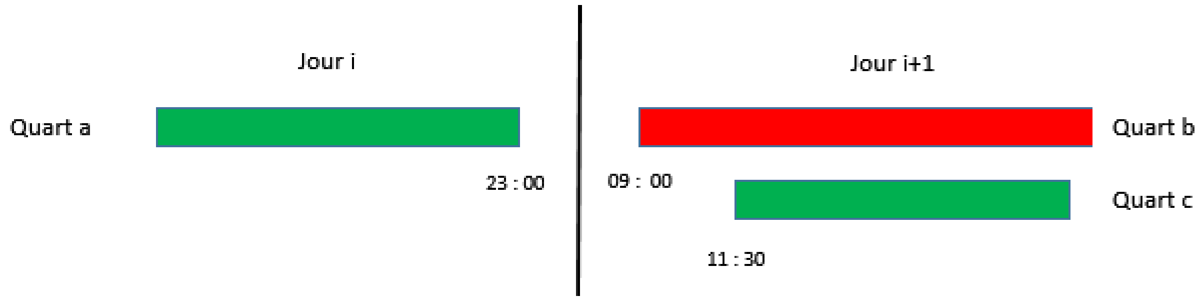

Figure 3.2 Repos

Nous pouvons, pour exprimer cette contrainte, déclarer qu'en retranchant l'heure de fin de quart du jour i à l'heure de début de quart du jour $i+1$, nous devons obtenir au moins d_{min} heures. Soient Y_a, Y_b et Y_c des variables binaires correspondant au choix de sélectionner ou non chacun des quarts a, b et c. Soient également f_a, d_b et d_c les heures de fin et de début des quarts a, b et c. Nous pouvons écrire :

$$f_a Y_a + d_{min} \leq d_b Y_b + d_c Y_c \tag{3.1}$$

Cette façon de procéder n'est pas parfaite. En effet, ajouter cette contrainte au moment de l'optimisation pourrait poser quelques problèmes de résolution si la relaxation linéaire propose des choix tels que : $Y_a = 1$ et $Y_b = Y_c = 0.5$. De plus, pour un problème beaucoup plus gros, i.e. avec beaucoup plus de quarts à considérer, les incidents causées par la relaxation linéaire pourraient être trop importants.

C'est pourquoi nous préférons savoir à l'avance quels quarts sont compatibles pour un même employé. Ainsi nous énumérons des ensembles $\mathcal{S}_e^h$ qui déterminent les quarts potentiellement attribuables à l'employé $e \in \mathcal{E}$ à la date $h \in \mathcal{H}$, tout en respectant les contraintes diverses du problème liées aux quarts : repos minimum, mais aussi nombre de jours travaillés par semaine.

3.1.3 Masse salariale

Il existe plusieurs façons de gérer la masse salariale. En fait, tout dépend de la façon dont nous voulons construire l'horaire. Si nous voulons instaurer une équité entre les employés, nous aurons tendance à utiliser une fonction de salaire croissante, constante par morceaux.

Cela coûte cher d'avoir un écart-type très grand concernant le nombre d'heures travaillées par les employés. Ainsi, un peu à la manière des heures supplémentaires payées plus cher, l'entreprise serait perdante si nous faisions travailler beaucoup plus un employé plutôt qu'un autre. L'autre façon de faire est d'instaurer un salaire horaire unique indépendant du nombre d'heures travaillées, dont le coût associé est défini par c. Dès lors, il n'y aucune raison qu'il y ait une quelconque équité entre les employés.

Pour ce qui nous concerne, nous pourrions tester les deux, mais la logique voudrait que nous privilégions la seconde option. En effet, notre modèle voulant favoriser les employés plus seniors au détriment des plus juniors, il n'y a aucune raison que nous cherchions à respecter une quelconque équité. S'ajoute à cela le fait qu'à poste égal, les employés les plus seniors sont en général les mieux payés dans une entreprise. Comme nous voulons favoriser le temps de travail des seniors, nous n'avons aucun intérêt à considérer des salaires inconstants.

3.2 Modèles

Nous présentons en premier lieu un modèle qui ne tient pas compte de la préférence avant d'ajouter cette considération.

3.2.1 Modèle de base

Une semaine avant l'horizon d'étude, le gérant du personnel doit préparer les horaires des employés et donc attribuer les quarts aux employés de sorte à couvrir la demande en tout temps pour assurer un service de qualité à la clientèle. Il faut remplir trois conditions pour qu'un quart s appartienne à $\mathcal{S}_e^h$:

- b_s doit être une période du jour h et une période de début admissible pour l'employé ;
- l_s correspond à une longueur de quart réalisable (par exemple, nous ne voulons pas de quarts d'une durée supérieure à 48 périodes) ;
- l'employé e doit être qualifié pour l'activité w_s et disponible le jour h.

Évidemment, les disponibilités et qualifications sont très variables d'un employé à un autre. De même, l'horaire d'un employé est réalisable s'il contient :

- au moins n^O jours de repos ;
- au plus un quart de travail par jour ;
- au moins n^R périodes de repos entre deux jours de travail consécutifs ;
- au plus n^P périodes de travail dans la semaine.

Il faut noter que pour des horizons plus longs qu'une semaine, il faut considérer d'autres conditions sur la répartition des jours de congé, et le nombre maximum de périodes travaillées doit être imposé sur chaque semaine, et non sur tout l'horizon. Ces informations étant données, nous allons pouvoir expliciter le problème de façon formelle et générique. Voyons d'abord la notation additionnelle.

Soient les variables :

— X_{es}^h une variable binaire qui vaut 1 si le quart $s \in S_e^h$ est affecté à l'employé $e \in E$ le jour h et 0 sinon ;

— O_e^h une variable binaire qui vaut 1 si un repos est assigné à l'employé $e \in E$ à la date $h \in H$ et 0 sinon ;

— L_e une variable entière spécifiant le nombre de périodes travaillées par e ;

— Y_s une variable entière indiquant le nombre de quarts anonymes $s \in \mathcal{S}^{\mathcal{A}}$ utilisés dans la solution ;

— U_k^p une variable entière indiquant le nombre de quarts anonymes utilisés à la période $p \in \mathcal{P}$ qui ont lieu sur le palier $k \in \mathcal{K}^U$;

— V_w^{kp} une variable entière indiquant le nombre de sur-couvertures pour l'activité $w \in \mathcal{W}$ sur la période $p \in \mathcal{P}$ qui ont lieu sur le palier $k \in \mathcal{K}^V$.

Soient :

— a_{sw}^p vaut 1 si le quart $s \in \mathcal{S}^A \bigcup\limits_{h\in\mathcal{H},e\in\mathcal{E}} \mathcal{S}_e^h$ couvre l'activité w durant la période p et 0 sinon ;

— M est une constante suffisamment grande ;

Le problème de construction d'horaires des employés peut se modéliser comme le programme en nombre entiers suivant :

$$Min \quad \sum_{e\in\mathcal{E}} cL_e + \sum_{s\in\mathcal{S}^A} c_s^Y l_s Y_s + \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}^U} c_k^U U_k^p + \sum_{p\in\mathcal{P}} \sum_{w\in\mathcal{W}} \sum_{k\in\mathcal{K}^V} c_k^V V_w^{kp} \tag{3.2}$$

Sujet à :

$$\sum_{s\in\mathcal{S}_e^h} X_{es}^h + O_e^h = 1, \qquad \forall e \in \mathcal{E}, h \in \mathcal{H} \tag{3.3}$$

$$\sum_{h\in\mathcal{H}} O_e^h \geq n^O, \qquad \forall e \in \mathcal{E} \tag{3.4}$$

$$\sum_{s\in\mathcal{S}^A} a_{sw}^p Y_s = \sum_{k\in\mathcal{K}^U} U_k^p, \qquad \forall p \in \mathcal{P}, w \in \mathcal{W} \tag{3.5}$$

$$MO_e^{h+1} + \sum_{s \in \mathcal{S}_e^{h+1}} b_s X_{es}^{h+1} - \sum_{s \in \mathcal{S}_e^h} (b_s + l_s + n^R) X_{es}^h \geq 0, \qquad \forall e \in \mathcal{E}, h \in \mathcal{H} \backslash \{7\} \tag{3.6}$$

$$\sum_{e \in \mathcal{E}} \sum_{h \in \mathcal{H}} \sum_{s \in \mathcal{S}_e^h} a_{sw}^p X_{es}^h + \sum_{s \in \mathcal{S}^A} a_{sw}^p Y_s - \sum_{k \in \mathcal{K}^V} V_w^{kp} = d_w^p, \qquad \forall p \in \mathcal{P}, w \in \mathcal{W} \tag{3.7}$$

$$X_{es}^h \in \{0, 1\}, \qquad \forall e \in \mathcal{E}, h \in \mathcal{H}, s \in \mathcal{S}_e^h \tag{3.8}$$

$$O_e^h \in \{0, 1\}, \qquad \forall e \in \mathcal{E}, h \in \mathcal{H} \tag{3.9}$$

$$L_e \in [0, n^P], \qquad \text{entière } \forall e \in \mathcal{E} \tag{3.10}$$

$$Y_s \geq 0, \qquad \text{entière } \forall s \in \mathcal{S}^A \tag{3.11}$$

$$V_w^{kp} \in [0, n_k^V], \qquad \text{entière } \forall w \in \mathcal{W}, p \in \mathcal{P}, k \in \mathcal{K}^V \tag{3.12}$$

$$U_k^p \in [0, n_k^U], \qquad \text{entière } \forall p \in \mathcal{P}, k \in \mathcal{K}^U \tag{3.13}$$

La fonction objectif (3.2) vise à minimiser les coûts engendrés par l'affectation des quarts aux employés, l'affectation des quarts anonymes ainsi que la sur-couverture. Les contraintes (3.3) et (3.4) représentent respectivement les contraintes d'affectation des quarts de travail aux employés et l'affectation des repos aux employés. La contrainte (3.5) permet de calculer les variables U_k^p. Le repos minimum entre deux quarts consécutifs pour un employé est contrôlé par la contrainte (3.6). Pour chaque période et chaque activité, la contrainte (3.7) permet de garantir que la demande sera couverte par un nombre suffisant d'employés, ou à défaut, des quarts anonymes. Enfin, les domaines associés aux variables de décision sont régis par les contraintes (3.8) à (3.13). Remarquons que pour les contraintes (3.5) et (3.7), la variable d'un palier k va être égale à 0 sauf si la variable du palier précédent est à sa borne supérieure, et ceci à cause des coûts croissants.

3.2.2 Maximisation du nombre d'heures travaillées pour les employés seniors

Dans cette section et la suivante, nous entrons dans la partie principale de ce mémoire. Il s'agit d'intégrer la satisfaction de préférences au modèle précédemment présenté. En l'occurrence, nous nous occupons d'une seule préférence : nous cherchons à ce que les employés les plus seniors travaillent le plus possible. Ainsi, nous ferons en sorte qu'un employé junior ne puisse pas travailler plus qu'un employé qui est plus ancien, au moins à qualifications et disponibilités égales.

Pour ce faire, nous proposons de modifier la fonction objectif sans modifier les contraintes du problème initial. Un terme doit être ajouté à la fonction objectif qui résonne comme un bonus associé au fait qu'un employé travaille. Nous multiplions le nombre d'heures travaillées par un poids qui est un paramètre dépendant de l'ancienneté de l'employé. Aussi, nous proposons de résoudre le problème de façon séquentielle, donc notre approche ne sera pas exacte. En traitant le problème de cette façon, et en considérant les employés les plus seniors en premier, nous favorisons le respect des préférences. Ainsi, notre nouvelle fonction objectif prend du sens. Le modèle s'énonce comme suit :

$$Min \quad \sum_{e\in\mathcal{E}} cL_e + \sum_{s\in\mathcal{S}^A} c_s^Y l_s Y_s + \sum_{p\in\mathcal{P}} \sum_{k\in\mathcal{K}^U} c_k^U U_k^p + \sum_{p\in\mathcal{P}} \sum_{w\in\mathcal{W}} \sum_{k\in\mathcal{K}^V} c_k^V V_w^{kp} - \sum_{h\in\mathcal{H}} \sum_{s\in\mathcal{S}_e^h} \sum_{e\in\mathcal{E}} \lambda_e \gamma_{es}^h X_{es}^h \tag{3.14}$$

Sujet aux contraintes (3.3) à (3.13)

Dans ce modèle, γ_{es}^h représente le nombre d'heures travaillées par l'employé e durant le quart s à la date h et λ_e est positif et correspond à un poids représentant l'ancienneté de chaque employé. Les poids sont décroissants en fonction de la seniorité. Ce sont des paramètres que nous fixons en amont de l'optimisation. Le bonus est négatif car nous cherchons à minimiser la fonction objectif. Donc favoriser un employé plutôt qu'un autre doit permettre de diminuer le coût de la solution.

3.3 Méthode de résolution de base

En pratique, le modèle est trop grand pour le résoudre de manière exacte tout en respectant les contraintes de temps de calcul. Ainsi l'approche de résolution suivante est heuristique, dans le but d'obtenir des temps de calcul assez rapides. La méthode de résolution initiale que nous avons choisie se base évidemment sur le modèle énoncé précédemment. Cette approche est séquentielle, car elle s'effectue par groupe d'employés. Nous entrons en paramètre un entier désignant la taille de ces groupes. Dans la plupart des résultats présentés, ce paramètre est initialisé par défaut à 5, de sorte à avoir 5 à 10 groupes à traiter. Dans le cas contraire, nous préciserons à chaque fois quelle valeur a été donnée à ce paramètre. La variation de ce paramètre sera l'objet d'une étude présentée dans le chapitre 5. Bien sûr, dans le cas où l'instance ne comporte pas un nombre d'employés qui est un multiple de 5, le dernier groupe peut être plus petit.

En ce qui concerne la première itération du processus, elle consiste simplement à considérer les 5 employés les plus seniors. Les autres n'entrent pas en compte. Dans le fichier d'entrée, nous fournissons également le paramètre λ_e pour chaque employé du groupe. Nous utilisons des paramètres décroissants pour imposer la seniorité au sein du groupe. Ceux-ci feront également l'objet d'une analyse de sensibilité car ils influent de façon non négligeable sur les solutions obtenues. L'optimisation essaie de combler la demande au mieux avec les employés du problème, et satisfait tout le reste de la demande avec des quarts anonymes. Pour terminer la première itération, l'horaire des employés du groupe courant est fixé. Cela signifie que les variables pour les quarts attribués aux employés du groupe sont fixées à 1. De même les variables des quarts non attribuées et associées à ces employés sont toutes fixées à 0. De plus, les courbes de demande sont mises à jour.

La deuxième itération prend en compte cette mise à jour et les horaires fixés, pour optimiser l'horaire des employés les plus seniors suivants. De même si la demande n'est pas comblée à l'issue de l'optimisation, on la respecte grâce à l'ajout de quarts anonymes. Puis le processus itératif poursuit jusqu'au dernier groupe. À la fin de la dernière itération, l'optimiseur comble une dernière fois les restes de demande avec des quarts anonymes.

L'avantage majeur de cette approche est que nous favorisons le respect de la préférence dont il est question. En effet, comme il y a toute la demande à couvrir à la première itération et qu'il n'y a pas assez d'employés pour la couvrir, nous assurons que ces employés vont travailler le plus possible. Le respect de la seniorité à l'intérieur du groupe dépend des poids λ_e donnés aux employés du groupe, comme de la taille des groupes.

3.4 Évaluation des solutions obtenues

Il faut remarquer que le respect parfait de la seniorité ne peut se faire qu'à compétences et disponibilités égales. Dans ce cas-là, nous observons que l'employé le plus ancien est bien celui qui travaille le plus (et qu'il travaille le plus possible) mais aussi que le plus junior est celui qui travaille le moins. Néanmoins, il nous faut trouver un moyen de pouvoir s'assurer que la solution obtenue fournit un résultat cohérent d'un point de vue du respect des préférences, lorsque les disponibilités et qualifications sont multiples et variées.

Par exemple, si l'employé le plus senior n'a que 4 heures de disponibilité sur l'horizon d'étude, il faut s'assurer que nous lui donnons bien ces 4 heures. En revanche, nous ne pouvons pas forcer tous les employés plus juniors à travailler moins de 4 heures dans la semaine sous peine de ne pas satisfaire la demande. C'est pourquoi nous avons implémenté un évaluateur de solutions. Il permet de s'assurer que nous avons fait de notre mieux pour donner le maximum d'heures possible à chaque employé en évaluant un ratio. Ce ratio est calculé selon l'algorithme suivant.

Algorithme 1 Évaluation des solutions

Soient : h_e l'horaire de l'employé e dans la solution fournie obtenu avec la méthode de résolution de base, $T(h)$ le temps travaillé dans l'horaire h d'un employé et k une constante positive

pour tout employé $e \in \mathcal{E}$ **faire**

  $\lambda_e = k$ et $\lambda_{e'} = 0$ pour $e' \in \mathcal{E} \setminus \{e\}$

  Résoudre le problème avec la méthode de base. On obtient l'horaire H_e pour e

  Calculer le ratio : $ratio_e = T(h_e)/T(H_e)$

  Fixer l'horaire h_e de l'employé e selon la méthode de base

fin pour

L'Algorithme 1 consiste foncièrement à traiter le problème de la même manière que celui du *PBS*. Ce problème a été résolu maintes fois dans le cadre de la fabrication d'horaires de personnel de bord dans les compagnies aériennes. Dans l'ensemble $\mathcal{E}$, ordonné de l'employé le plus senior au plus junior et pour une constante k positive, on choisit l'employé numéro 1. À l'aide d'un solveur commercial de programmation en nombre entiers, on optimise son horaire en lui attribuant un bonus égal à k et en complétant la demande avec des quarts anonymes. Une fois cet horaire trouvé, on fixe les quarts de l'employé e issus de la solution d'origine, c'est à dire que nous fixons à 1 les variables des quarts de l'employé e dans l'horaire h_e et à 0 les autres variables de cet employé. Ensuite, on choisi l'employé numéro 2 et on procède de la même manière. L'approche utilisée a pour avantage de fournir l'horaire optimal à chaque itération, compte tenu de ce qui a été fixé précédemment. Ainsi le premier employé aura son horaire optimal. Le dernier aussi selon ce qui a été fixé précédemment. Cela est très utile pour nous assurer que chaque employé s'est bien vu attribuer le maximum d'heures possible, et plus particulièrement les plus seniors. C'est ainsi que nous résolvons le problème autant de fois qu'il y a d'employés. Cela nous informe sur combien d'heures chaque employé pourrait travailler dans le meilleur des cas. Il s'agit alors de comparer la valeur obtenue à celle de la solution découlant du problème initial dans lequel nous traitons des groupes de 5 employés à la fois. On en déduit un ratio qui nous renseigne sur la qualité de notre solution initiale en termes de satisfaction de la préférence.

En effet, un ratio égal à 1 signifie que nous avons respecté au mieux le nombre d'heures que l'employé pouvait travailler selon les horaires fixés pour les employés plus seniors. Un ratio inférieur à 1 signifie que nous aurions pu mieux contenter l'employé. Enfin, il est possible qu'il y ait des ratios supérieurs à 1 dans le cas où de la sur-couverture apparaîtrait. Celle-ci coûte très cher dans la solution, néanmoins avec le modèle proposé, il est possible de la forcer. Par exemple, si le poids λ_e est très élevé, le bonus associé à l'employé qui travaillera en sur-couverture compensera le coût associé à celle-ci. Par conséquent, la solution sera "meilleure" au sens de l'optimiseur, avec de la sur-couverture.

Lorsque nous évaluons une solution, nous résolvons le problème pour chaque employé. Mais il est très important de signaler que nous prenons en compte l'horaire obtenu dans la solution d'origine à la fois pour faire la comparaison, mais aussi pour obtenir le nombre d'heures attribuables maximum. Cela signifie que lorsque nous évaluons le premier employé, nous résolvons normalement. Pour le deuxième, nous fixons non pas l'horaire du premier employé trouvé par l'évaluateur, mais bien l'horaire du premier employé dans la solution d'origine. De même lorsque l'évaluation est effectuée pour le troisième employé, les horaires fixés pour les employés 1 et 2 sont ceux de la solution d'origine.

Il est important de souligner que cette partie de la résolution n'entre pas dans le processus opérationnel de fabrication des horaires. En fait, elle nous sert à vérifier que l'approche de résolution envisagée permet bel et bien de satisfaire la préférence choisie.

3.5 Présentation des jeux de données

Avant tout, il convient de rappeler les contours de notre étude. L'objectif est de satisfaire les préférences des employés les plus seniors, tout en minimisant les coûts. Les temps de calcul ne doivent pas dépasser la dizaine de minutes idéalement. Aussi, les instances sont de taille moyenne. Ceci implique que nous n'excédons jamais les 50 employés. De même nous ne descendons pas en dessous de 25 employés, mais nous avons pour vocation de proposer une méthode qui pourrait satisfaire des instances de plus grande taille. C'est ainsi que nous travaillons sur trois jeux de données différents. Un comportant 25 employés, un deuxième de 32 employés et enfin un de 47 employés. Toutes ces données ont été fournies par notre partenaire industriel. Elles ont pu être modifiées pour les besoins de l'étude.

En effet, l'instance à 25 employés a, par exemple, été profondément modifiée pour servir notre propos. Nous nous sommes exemptés des contraintes liées à la disponibilité des employés. Autrement dit, tous les employés sont rendus disponibles en tout temps sur l'horizon de l'étude, ce qui n'est pas réaliste. Effectivement, certains employés peuvent ne pas travailler un certain jour, ou toujours finir avant 16 heures par exemple. Aussi, nous avons conservé les qualifications originelles des employés. Le temps de travail est limité à 5 jours par semaine et 40 heures par semaine. L'objectif de cette instance est de montrer facilement que nous arrivons à satisfaire les préférences des employés les plus seniors.

Pour ce qui est de l'instance avec 32 employés, il n'y a pas eu de modification majeure, comme pour celle avec 47 employés. Ces instances servent à montrer que notre modélisation et nos algorithmes fonctionnent dans des cas réels, c'est-à-dire les cas où il n'est pas possible d'observer immédiatement que nous avons été capables de satisfaire la préférence des employés. Par exemple, dans l'instance comprenant 32 employés, l'employé le plus senior ne travaille jamais, mais cela est dû au fait qu'il n'est pas disponible sur l'horizon. Par conséquent, en regardant les résultats brutalement, nous pourrions dire que ses préférences ne sont pas satisfaites. Il est à noter que dans tous les jeux de données, nous définissons arbitrairement le niveau de seniorité de chaque employé. Le tableau 3.3 résume les caractéristiques principales des diverses instances. Dans la prochaine section, nous abordons les résultats obtenus pour l'approche considérée. Tout ce qui suit concernera les trois jeux de données explicités ci-dessus.

Tableau 3.3 Présentation des instances utilisées

Nombre d'employés	25	32	47
Nombre d'activités	5	3	5
Demande totale	938	945	1831
Durée maximum des quarts	8h00	10h00	10h00
Durée minimale des quarts	3h00	3h00	4h00
Nombre de jours maximum par semaine	5	6	6

3.6 Résultats

Dans cette partie, nous exposons deux types de résultats. À la fois les résultats témoins, fruits d'une optimisation globale sans maximisation du nombre d'heures travaillées, mais aussi les résultats issus de notre approche. Tous les coûts dont nous parlons dans le reste du mémoire ne tiennent pas compte des bonus, car ceux-ci sont purement fictifs pour augmenter le nombre d'heures travaillées. Dans toute la suite, quel que soit l'algorithme ou l'optimisation réalisé, tous les programmes en nombres entiers sont résolus à partir du solveur commercial XPress-MP, version 7.9. Nous utilisons les paramètres par défaut à l'exception de la tolérance sur le saut d'optimalité, qui a lui été fixé à zéro pour assurer l'optimalité de la solution. Les ordinateurs utilisés ont comme système d'exploitation Windows 7, et un processeur Intel (R) Core (TM) i7-3770 à 3,40GHz.

3.6.1 Résultats sans préférence

Pour commencer, nous allons présenter les résultats sans prendre en compte la préférence. Cela va nous servir dans plusieurs optiques. Déjà, nous allons pouvoir regarder de façon brute le nombre d'heures travaillées par les employés dans ce cas précis. D'autre part, nous allons observer quelle est la solution optimale en termes de coût, de nombre total d'heures travaillées, de nombre total d'heures de sur-couverture et de nombre total d'heures de sous-couverture. Cela nous permettra de mettre en lumière les points positifs ainsi que les points négatifs de notre méthode. Au niveau méthodologique, cela revient à résoudre le problème avec un seul groupe de taille égale au nombre d'employés dans le jeu de données. De plus nous considérons des bonus λ_e égaux à 0, pour tous les employés.

Maintenant, nous allons présenter les résultats pour chaque instance grâce à la figure 3.3 et au tableau 3.4. Ce dernier récapitule les points importants à retenir et qui serviront de comparaison pour la suite des résultats. En effet, nous entendons grandement améliorer certains points, notamment la satisfaction moyenne des préférences. Cet item, est le fruit de l'évaluation de la solution. À l'inverse, nous allons garder aussi en mémoire le coût des solutions, les temps liés aux quarts anonymes et aux sur-couvertures en espérant ne pas trop les dégrader. Remarquons également que les temps de calcul sont plus élevés pour l'instance de 25 employés. Cela est dû au fait que les disponibilités y sont totales, donc les ensembles de quarts disponibles pour les employés sont plus grands. De ce fait le modèle contient de très nombreuses variables. Cette remarque vaut bien sûr pour le reste du mémoire.

Nous n'observons sur la figure 3.3 aucune décroissance définie. Cela est normal car il n'y a pas de préférence en jeu. Ainsi nous pourrons comparer la différence avec les résultats obtenus avec l'approche que nous avons considérée.

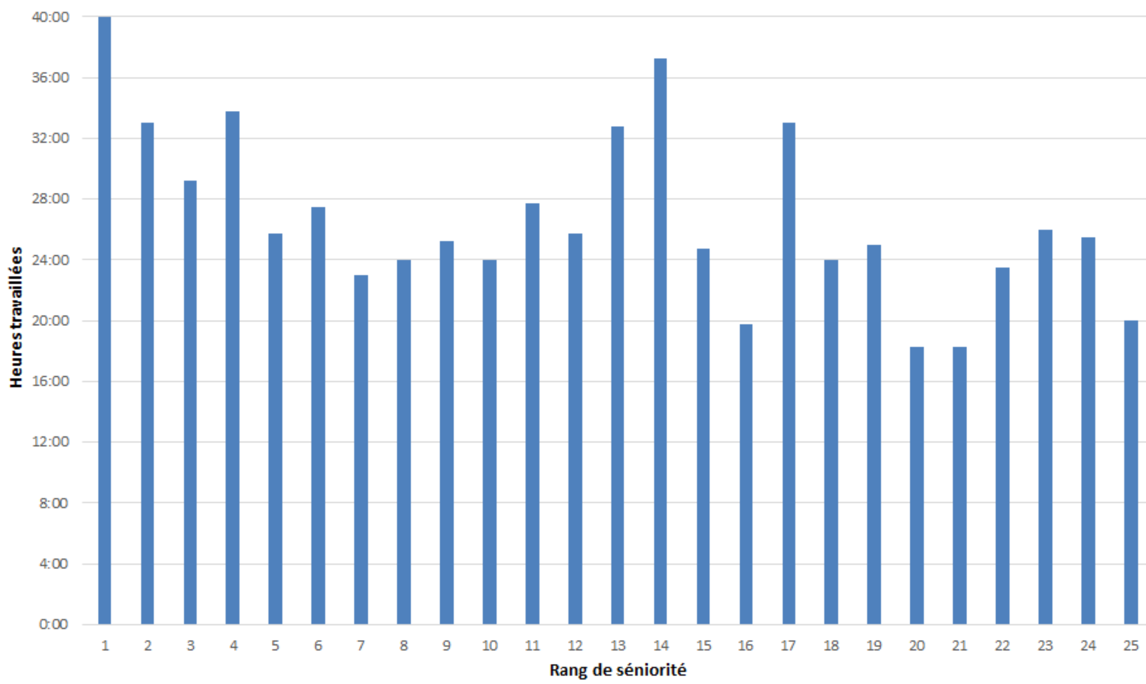

Figure 3.3 Nombre d'heures travaillées en fonction de la seniorité pour 25 employés sans la préférence

Tableau 3.4 Informations sur les tests sans préférence

Instance	25 employés	32 employés	47 employés
Nombre d'heures travaillées	667h00	633h30	1066h30
Nombre de quarts anonymes	22	29	77
Nombre d'heures de quarts anonymes	10h15	10h45	27h30
Nombre d'heures de sur-couverture	15h45	13h30	25h45
Coût de la solution	156 000	172 913	395 276
Temps de calcul (en s)	443	43	91
Ratio moyen	71.1%	72.%	72.5%

Au tableau 3.4, nous constatons que les quarts anonymes sont inévitables. De même, il est impossible d'éviter un minimum de sur-couverture pour obtenir la solution optimale. Cela est dû aux diverses qualifications et disponibilités des employés, de même qu'à la construction des quarts pour les employés. Désormais, nous connaissons les objectifs à atteindre pour énoncer que nos solutions sont bonnes. Le premier est d'obtenir une courbe de décroissance parfaite pour le jeu de données de 25 employés. À cela nous voulons ajouter une amélioration significative de la satisfaction des employés et plus particulièrement au niveau des plus seniors.

Accessoirement, nous cherchons à ne pas augmenter de manière trop significative (i.e. moins de 5%) le coût des solutions. Notons que la majorité de ces coûts ne sont pas réels, car ils correspondent à des pénalités pour la sous-couverture et la sur-couverture. Nous veillerons également à ne pas décupler les temps de calcul.

3.6.2 Résultats avec maximisation du nombre d'heures travaillées

Cette sous-partie a pour but d'exposer que nous sommes capables, avec la méthode de résolution mise en place, de satisfaire les préférences des employés. Nous montrons pour l'instance témoin (25 employés), que nous pouvons respecter la décroissance du nombre d'heures travaillées en fonction de la seniorité des employés. La figure 3.4 montre cette courbe pour l'instance de 25 employés résolue par groupes de 5 employés.

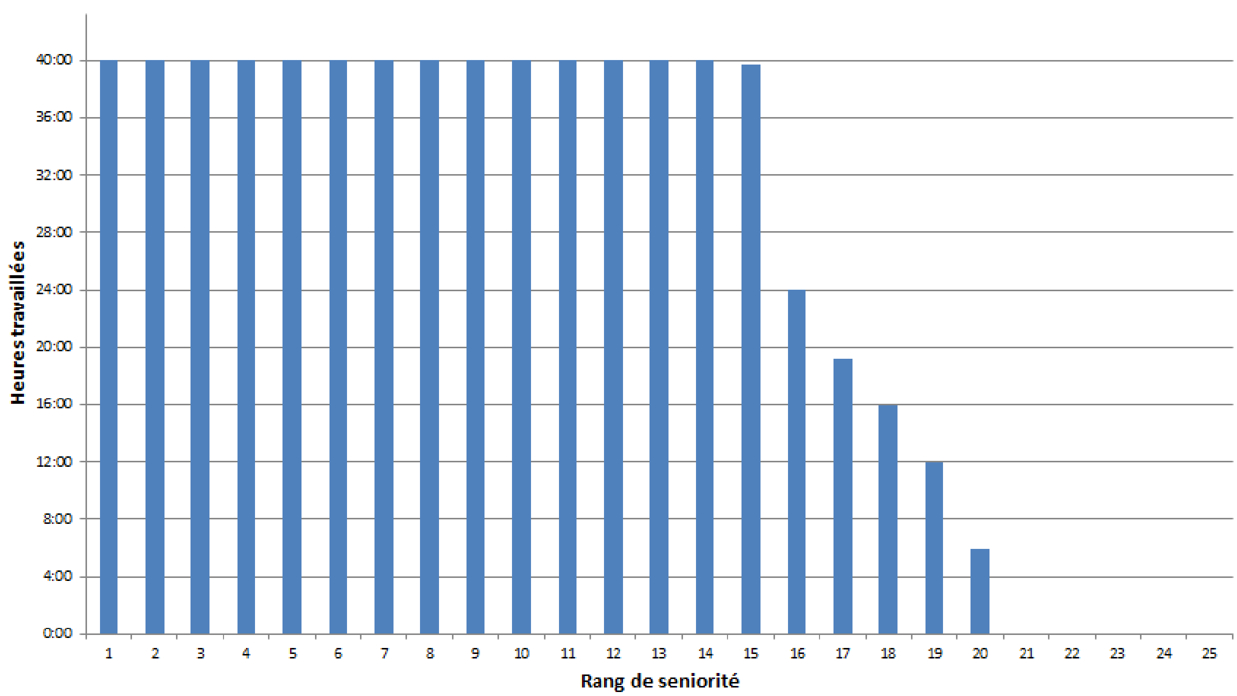

Figure 3.4 Nombre d'heures travaillées en fonction de la seniorité pour 25 employés avec la préférence

Le problème est que, pour les autres instances, l'algorithme ne peut produire une telle décroissance du nombre d'heures travaillées. Les qualifications, les disponibilités et les diverses règles introduites dans le fichier d'entrée traduisent en effet cette impossibilité. Par exemple, il suffit que l'employé le plus senior ne soit disponible que 4 heures sur l'horizon d'étude pour que nous soyons complètement assuré de ne pas obtenir une courbe de décroissance. Forcément, pour combler la demande, il faudra que les employés plus juniors travaillent plus d'heures, sans pour autant que les préférences n'aient pas été satisfaites. C'est pourquoi nous utilisons l'évaluateur décrit par l'algorithme 1. Bien qu'imparfait, il nous permet d'avoir une idée assez précise de la satisfaction de chaque employé.

Les résultats montrent que notre méthode fonctionne assez bien pour les instances a priori complexes (32 et 47 employés). L'évaluateur révèle surtout que les employés parmi les 47% plus seniors sont toujours satisfaits à plus de 95% (voir annexe A). Le tableau suivant permet de voir la satisfaction moyenne des employés selon les différentes instances.

Tableau 3.5 Ratio moyen en fonction de l'instance et de la taille des groupes

Instance	25 employés	32 employés	47 employés
Groupes de 3 employés	99.0%	96.1%	98.6%
Groupes de 5 employés	99.1%	93.1%	96.9%
Groupes de 8 employés	95.7%	95.1%	96.1%

Ces moyennes étant inférieures à 1, il apparaît clair qu'il y a des employés pour lesquels la préférence n'est pas pleinement satisfaite. De façon très générale, on observe que l'optimisation parvient bien à satisfaire les préférences des employés les plus seniors. Cela est confirmé à la fois par l'instance de 25 employés, mais aussi suite à l'évaluation pour les instances de 32 et 47 employés. Néanmoins, il est aisé de constater que cela se fait au détriment des coûts des solutions. En effet, la comparaison avec la résolution des instances "sans préférence" permet d'observer qu'on accroît de façon considérable les coûts pour l'entreprise (voir tableau 3.6). Bien sûr, ceci n'est absolument pas acceptable pour l'entreprise cliente. C'est pourquoi nous avons envisagé d'autres approches pour l'optimisation initiale, qui seront présentées dans le chapitre suivant.

De façon pragmatique, nous pouvons constater que les coûts deviennent démesurément grands dès lors que nous cherchons à satisfaire la préférence. L'origine des coûts élevés est la sur-couverture, comme la sous-couverture. Il est aisé de constater que le nombre de quarts anonymes et le nombre d'heures de sur-couverture augmentent considérablement avec cette approche. Aussi, on remarque que le nombre d'heures travaillées est plus faible en général pour les instances résolues avec préférence. Cela traduit un problème dans le processus d'attribu-

Tableau 3.6 Comparaison des résultats avec et sans la préférence (SP)

Nombre d'employés	25 SP	25	32 SP	32	47 SP	47
Nombre d'heures travaillées	667h00	**677h00**	633h30	**561h45**	1066h30	**899h30**
Nombre de quarts anonymes	22	**47**	29	**65**	77	**127**
Nombre d'heures quarts anonymes	10h15	**21h00**	10h45	**86h30**	27h30	**239h45**
Nombre d'heures de sur-couverture	15h45	**36h30**	13h30	**29h30**	25h45	**56h45**
Coût de la solution	156 000	**345 000**	172 913	**755 430**	395 276	**1 857 864**

tion des quarts. Dans notre approche d'origine, nous fixons les horaires d'un groupe avant de passer au groupe suivant. Entre temps, la demande a été comblée avec des quarts anonymes couteux et ceux-ci n'ont jamais pu être retirés.

Un exemple de ce phénomène a été fourni dans l'annexe B. Nous remarquons bien le problème en observant la demande de l'activité 14765. Celle-ci est constante égale à 1 chaque jour de la semaine. Or, cette activité ne peut être réalisée que par 8 employés différents. Leur niveau d'ancienneté est assez important (ils sont dans les 3 premiers groupes traités). Il faut aussi signaler que ces employés sont aussi qualifiés pour d'autres activités. Ainsi, lors de l'optimisation séquentielle, il se trouve que les employés qualifiés pour l'activité 14765 ne se la voient pas attribuée. La demande est comblée par des quarts anonymes et nous passons au groupe suivant. Ensuite, le problème est qu'aucun employé n'est qualifié pour cette activité. Ainsi, nous nous retrouvons avec de nombreux et longs quarts anonymes qui coûtent très cher. De plus, nous augmentons la sur-couverture sur les activités auxquelles les employés seniors ont été affectés, à cause des bonus associés aux plus juniors qui ne sont qualifiés que pour cette activité (14768).

À la lumière de ces résultats, nous avons été poussés à adopter d'autres approches pour réparer les coûts de nos solutions, tout en gardant un niveau acceptable de satisfaction des préférences.

Conclusion sur cette première approche Même si les résultats obtenus avec cette première approche ne sont pas acceptables d'un point de vue opérationnel, elle a le mérite de nous montrer quelles sont les solutions optimales au sens de la maximisation des heures travaillées par les employés les plus seniors. Par là, nous suggérons que les prochaines approches auront pour but de se rapprocher des coûts engendrés par les solutions sans préférence, tout en se rapprochant de la satisfaction des employés seniors dans les solutions avec préférences. Il apparaît clair que nous pouvons de toute façon améliorer les solutions actuelles rien qu'en optimisant l'attribution des activités aux employés seniors.

CHAPITRE 4 AUTRES APPROCHES DE RÉSOLUTION ET STRATÉGIES DE RÉDUCTION DES TEMPS DE CALCUL

Comme cela a été abordé dans le chapitre précédent, il convient de trouver des solutions plus acceptables en contexte opérationnel. En effet, aucune entreprise n'acceptera de multiplier le coût des horaires par deux sous prétexte que la seniorité est respectée. Ainsi, nous avons élaboré deux méthodes proches de celle initiale, pour trouver le meilleur compromis entre coût, satisfaction et temps de calcul. En premier lieu, nous exposons en détail ces méthodes, leur genèse, et l'inconvénient auquel elles répondent. Puis nous exposerons les résultats obtenus avec ces méthodes. Ceux-ci nous permettront d'évaluer laquelle des trois approches envisagées est la meilleure.

De plus, nous avons tenté d'élaborer des stratégies de réduction des temps de calcul. Le but serait d'améliorer les temps de calcul sans détériorer les solutions.

4.1 Autres approches de résolution

Pour résoudre le problème initial, l'approche proposée n'est pas unique. En effet, dans la mesure où ce travail est heuristique, il existe une multitude d'approches différentes. Évidemment, toutes ne sont pas pertinentes, mais il est possible d'en extraire quelques unes qui peuvent se valoir. L'approche initiale proposée est très efficace pour satisfaire les préférences des employés, de façon indéniable. Néanmoins, elle a ses limites. En effet comme l'ont montré les résultats du chapitre précédent, les coûts associés à ces solutions sont exorbitants. Particulièrement, la répartition des activités n'est pas optimale et crée potentiellement beaucoup de sous-couverture et de sur-couverture. Ce constat nous pousse à envisager d'autres façons de procéder pour estimer quel serait l'impact de diverses modifications, en espérant trouver un bon équilibre entre :

— la satisfaction des préférences qui est effective avec l'approche de départ.

— le nombre de quarts anonymes et de sur-couverture qui sont bien trop nombreux dans cette approche.

— le temps de calcul, qui est une contrainte importante du cahier des charges.

4.1.1 Évaluateur

L'évaluateur tel qu'il a été défini dans le chapitre précédent ne peut plus être utilisé pour évaluer les solutions des nouvelles approches. Ainsi, celui-ci subit aussi un changement pour s'adapter à chaque approche. Comme précédemment, il compare le nombre d'heures travaillées dans la solution de référence à une solution « idéale » qu'il calcule. Cette solution « idéale » est obtenue en utilisant la même méthode que celle de la solution de référence, pour obtenir une comparaison qui a du sens. De même qu'auparavant, l'évaluateur traitera les employés un par un, affectant le meilleur horaire possible à chaque itération.

4.1.2 Ajout d'une contrainte stricte

Le problème majeur rencontré par l'approche d'origine est la génération de très nombreux quarts anonymes et de nombreuses sur-couvertures. Comme indiqué, cela est majoritairement dû à la façon dont sont attribuées les activités. La cause de tout cela est le caractère séquentiel de notre idée, mêlé au fait que l'on comble la demande avec des quarts anonymes. Ainsi, pour les premiers groupes, l'optimiseur a le choix des activités à donner aux employés, peu importe les qualifications des juniors. Le but de l'approche suivante est de mieux répartir les activités entre les employés, tout en gardant un niveau de satisfaction élevé.

L'idée générale est la suivante. Nous ne fixons pas les horaires des employés seniors à la fin de la première itération (i.e. le premier groupe). Par contre, nous fixons le nombre d'heures travaillées par ces employés, et nous conservons les quarts anonymes pour combler la demande. Ainsi nous ajoutons une contrainte dure pour chaque employé du groupe traité. Cette contrainte s'exprime de la manière suivante :

$$\sum_{s \in \mathcal{S}_e^h} \gamma_{es}^h X_{es}^h \geq \gamma_e, \qquad \forall e \in \mathcal{E} \tag{4.1}$$

γ_e représente le nombre d'heures fixé, trouvé dans les itérations précédentes, pour l'employé e et γ_{es}^h représente le nombre d'heures attribuées à l'employé e au cours du quart s à la date h dans l'itération courante.

Cette contrainte indique que les employés traités dans les groupes précédents doivent au moins travailler autant dans l'itération courante. Cela induit deux choses :

— Premièrement, nous ne nous intéressons plus à un nombre défini d'employés à la fois. En effet, si la première itération comporte toujours $taille_{groupe}$ employés, la deuxième

itération en traite deux fois plus, etc. Et la dernière itération traite tous les employés en même temps considérant toutes les contraintes ajoutées au problème.
— Deuxièmement, les quarts pour les employés seniors ne sont pas fixés. Donc l'attribution des activités non plus. Ainsi, si un employé senior travaillait 35h dans la première itération avec 5 quarts sur l'activité 1, il pourra à la fin de l'optimisation avoir 5 quarts répartis sur les activités 2 et 3. L'important est qu'il aura travaillé au minimum 35h à la fin de la dernière itération.

Les avantages de cette méthode sont bien évidemment la réduction du nombre de quarts anonymes. Si on reprend l'exemple de l'annexe B, et bien on se rendra compte que les premiers employés auront des quarts ressemblant plus à ceux de la solution sans préférence. Une conséquence directe de la réduction du nombre de quarts anonymes est la réduction du nombre de sur-couvertures. En effet dans les difficultés que nous avons rencontrées, les deux sont assez corrélées. Les bonus attribués ont tendance à favoriser cette sur-couverture. Et les quarts anonymes étant le fruit de la répartition des activités, il est aisé de constater que la sur-couverture l'est également.

Il existe en contrepartie un inconvénient majeur possible : l'allongement significatif du temps de calcul. Effectivement, plus on avance dans les itérations, plus la taille du problème grandit. En plus de cela, on rajoute des contraintes au problème pour chaque employé traité.

En ce qui concerne la satisfaction des préférences, il est difficile de prédire ce qu'il va se passer. Néanmoins, il apparaît difficile de la dégrader en comparaison avec l'approche initiale, du moins pour les employés seniors. Étant donné que l'évaluateur fonctionne sur le système du *PBS*, et que le nombre d'heures global va augmenter, il paraît tout à fait improbable que les employés seniors n'aient pas un ratio de 1. Par contre, l'évaluateur étant basé sur l'approche utilisée, il est possible que le changement de méthode induise des changements sur la satisfaction des juniors.

4.1.3 Variables réelles pour les employés juniors

La façon de procéder que nous allons présenter a pour but de supprimer l'utilisation des quarts anonymes au profit des employés plus juniors. Nous utilisons donc des variables entières pour les employés du groupe courant, et nous utilisons des variables réelles incluses dans [0 , 1] pour les employés juniors. Les quarts anonymes sont toujours utilisés pour combler la demande, mais leur utilisation devrait être considérablement réduite par la prise en compte des employés juniors. L'approche séquentielle existe toujours dans la mesure où ce sont des groupes d'employés qui ont des variables entières.

Concrètement, l'optimisation se déroule comme suit :

- Nous optimisons pour le premier groupe (supposons de taille 5 et comprenant les employés d'indices 1 à 5) avec le modèle décrit dans le chapitre précédent. On minimise la fonction objectif (3.14), sujet aux contraintes (3.3), (3.4), (3.5), (3.6), (3.7), (3.10), (3.11), (3.12). De plus les contraintes (3.8) et (3.9) sont modifiées et remplacées par :

$$X_{es}^h \in \{0,1\}\,, \qquad \text{pour } e \in \{1,\ldots,5\} \subseteq E, h \in \mathcal{H}, s \in \mathcal{S}_e^h \tag{4.2}$$

$$X_{es}^h \in [0,1], \qquad \forall e \in \mathcal{E} \setminus \{1,\ldots,5\}, h \in \mathcal{H}, s \in \mathcal{S}_e^h \tag{4.3}$$

$$O_e^h \in \{0,1\}\,, \qquad \text{pour } e \in \{1,\ldots,5\} \subseteq E, \forall h \in \mathcal{H} \tag{4.4}$$

$$O_e^h \in [0,1], \qquad \forall e \in \mathcal{E} \setminus \{1,\ldots,5\}\,, \forall h \in \mathcal{H} \tag{4.5}$$

  Ces relations viennent souligner que les quarts de travail et les repos peuvent être attribuer de façon partielle aux employés plus juniors, dans le but d'améliorer la répartition des activités et le coût des solutions par conséquent.
- À la fin de cette optimisation, nous fixons les horaires des employés du groupe courant.
- Nous passons au groupe suivant et nous procédons de la même façon avec le problème restreint aux employés plus juniors.

Les avantages et inconvénients ressemblent de très près à ceux de l'approche ajoutant des contraintes strictes. En effet, les avantages majeurs sont la réduction des quarts anonymes et de la sur-couverture. Traiter tous les employés en même temps permet de trouver des meilleures solutions (en terme de coût) dès la première itération (premier groupe). Ainsi, lorsque l'on fixe les horaires des employés seniors, il y a des chances que l'on améliore grandement la solution par rapport à l'approche d'origine.

À l'inverse, les temps de calcul risquent d'être plus longs que ceux d'origine. Prendre en compte les employés juniors dès la première itération tend à élargir le problème, même si ils ne sont pas considérés en variables entières.

Quant à la satisfaction des préférences, elle a toutes les chances d'être aussi bonne que dans l'approche initiale, car le changement de méthode ne va pas avoir d'impact réel sur le nombre d'heures travaillées. Le seul réel impact sera sur l'attribution des activités.

4.2 Réduction du temps de calcul

Comme annoncé, l'inconvénient majeur des nouvelles approches risque d'être l'allongement du temps de calcul. C'est pourquoi nous avons pensé à des stratégies pour essayer d'améliorer cet aspect. Deux idées majeures sont issues de la réflexion. La première consiste à ajouter des coupes au modèle. La seconde consiste à relaxer certaines contraintes du problème.

4.2.1 Ajout de coupes

Cette idée concerne l'approche ajoutant des contraintes strictes pour les employés déjà traités. En effet, nous avons remarqué que les contraintes imposées restreignaient l'utilisation de certains quarts comme l'illustre l'exemple suivant.

Un employé travaille 35 heures dans la solution à une itération donnée. Il a le droit de travailler 5 jours dans la semaine. Lors de la ré-optimisation, on veut éliminer certains quarts inutiles proposés à cet employé. Ainsi, on sait que pour des quarts qui durent entre 2 et 8h, il ne peut avoir plus d'un quart d'une durée de 5h. A fortiori, il ne peut pas y avoir plus d'un quart de 4h45, 4h30,..., 2h. Donc le but de la coupe est de limiter l'utilisation de ces quarts dans les solutions fractionnaires. Une telle coupe pour un employé e s'écrit :

$$\sum_{h \in \mathcal{H}} \sum_{\substack{s \in \mathcal{S}_h^e \\ l_s \leq x}} X_{es}^h \leq m(x) \tag{4.6}$$

où $m(x)$ représente un nombre maximum de quarts d'une durée d'au plus x périodes.

Pour bien définir une coupe associée à un employé e et un $x \in \mathbb{N}$, il faut trouver la plus grande valeur entière de $m(x)$ qui satisfait l'inéquation suivante :

$$m(x) * x + (nb_{jours} - m(x)) * d_{max} \geq T_e \tag{4.7}$$

où nb_{jours} est le nombre de jours autorisés à travailler dans la semaine (5 dans l'exemple précédent), $T_e \in \mathbb{N}$ est le temps de travail en périodes de l'employé dans la solution trouvée (140 périodes de 15 minutes ou 35h dans l'exemple), et $d_{max} \in \mathbb{N}$ est la durée maximum d'un quart de travail (32 périodes dans l'exemple). Cette inéquation traduit que le nombre maximum de quarts de durée x périodes ou moins doit être inférieur au ratio entre le nombre maximum de périodes possibles dans la semaine moins le nombre de périodes que l'employé travaille et la durée maximum d'un quart moins la durée x donnée.

Puisque $x \leq d_{max} - 1$, cette équation revient à

$$m(x) \leq \lfloor \frac{(T_e - nb_{jours} * d_{max})}{(x - d_{max})} \rfloor \tag{4.8}$$

Comme $nb_{jours} * d_{max} \geq T_e$ et $d_{max} > x$, cela revient à :

$$m(x) \leq \lfloor \frac{(nb_{jours} * d_{max} - T_e)}{(d_{max} - x)} \rfloor \tag{4.9}$$

Pour récapituler, dans l'exemple précédent le calcul donne pour $x = 20$ périodes :

$$m(20) * 20 + (5 - m(20)) * 32 \geq 140 \tag{4.10}$$

Ce qui revient à

$$m(20) \leq \lfloor \frac{5}{3} \rfloor \tag{4.11}$$

Puisque m est entier, il ne peut y avoir au plus $m(20) = 1$ quart d'une durée de 5h ou moins dans l'horaire de cet employé.
Le même raisonnement pour x égal à 12 périodes donne $m(12) \leq \lfloor 1 \rfloor$ d'où $m(12) = 1$.
Le même raisonnement pour x égal à 8 périodes donne $m(8) \leq \lfloor \frac{5}{6} \rfloor$ d'où $m(8) = 0$.
Il est à noter que si $m = 0$, alors la coupe peut s'imposer en éliminant tout simplement les variables associées aux quarts impliqués dans cette coupe. Ainsi, il ne peut y avoir aucun quart de 2h pour cet employé. Donc on resserre la relaxation linéaire du modèle en ajoutant les coupes proposées. Dans l'exemple ci-haut, on peut alors enlever tous les quarts d'une durée strictement inférieure à 3 heures.

4.2.2 Relaxation de la contrainte de repos

Nous venons d'exposer une méthode pour réduire les temps de calcul pour l'approche 1, qui ajoute des contraintes strictes au problème. Il convenait de trouver une solution pour pallier ce problème pour l'approche 2, qui consiste à rendre les variables des employés juniors réelles. Les coupes ne fonctionnent pas pour cette approche dans la mesure où l'on fixe les horaires à chaque itération. Donc les employés seniors ne sont plus dans le problème à la deuxième itération.

La méthode que nous employons consiste alors à relaxer certaines contraintes du problème pour les juniors. Quitte à relaxer la contrainte d'intégrité, nous pouvons aussi par exemple relaxer la contrainte de repos minimum. À l'inverse, nous ne pouvons pas relaxer les contraintes de nombre de jours travaillés par semaine, de nombre d'heures travaillées par semaine, ou encore de nombre de quarts par jour.

Nous choisissons donc de relaxer la contrainte de repos minimum, en n'imposant aucune durée minimum entre deux quarts consécutifs. Cela a pour but de réduire la taille du modèle et par conséquent d'accélérer les temps de calcul.

4.2.3 Coûts réduits des quarts

Un autre moyen de réduire le temps de calcul est de retirer un nombre de quarts conséquent du problème. À chaque résolution de la relaxation linéaire du problème, nous pouvons obtenir les coûts réduits des quarts. La première analyse à mener est donc de vérifier quels sont les coûts réduits des quarts proposés, mais aussi de regarder quels quarts ont été choisis dans la solution. Les figures 4.1 et 4.2 montrent la distribution des coûts réduits sur différentes instances et itérations.

En amont de l'optimisation, nous nous permettons de retirer tous les quarts anonymes dont le coût réduit est plus élevé qu'un paramètre v que l'on fixe à n'importe quelle valeur souhaitée. Par exemple, pour $v = 50$, nous obtenons une diminution de 67% des quarts proposés pour la figure 4.1 et de 85% pour la figure 4.2.

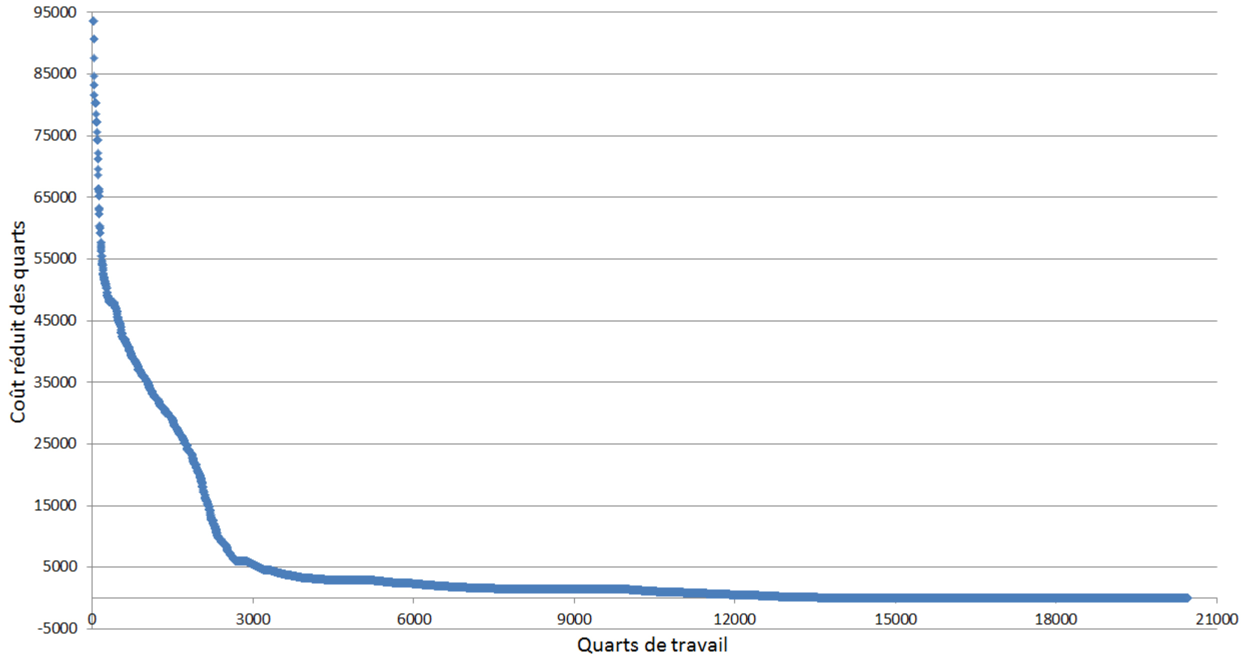

Figure 4.1 Distribution des coûts réduits (32 employés, groupe 1)

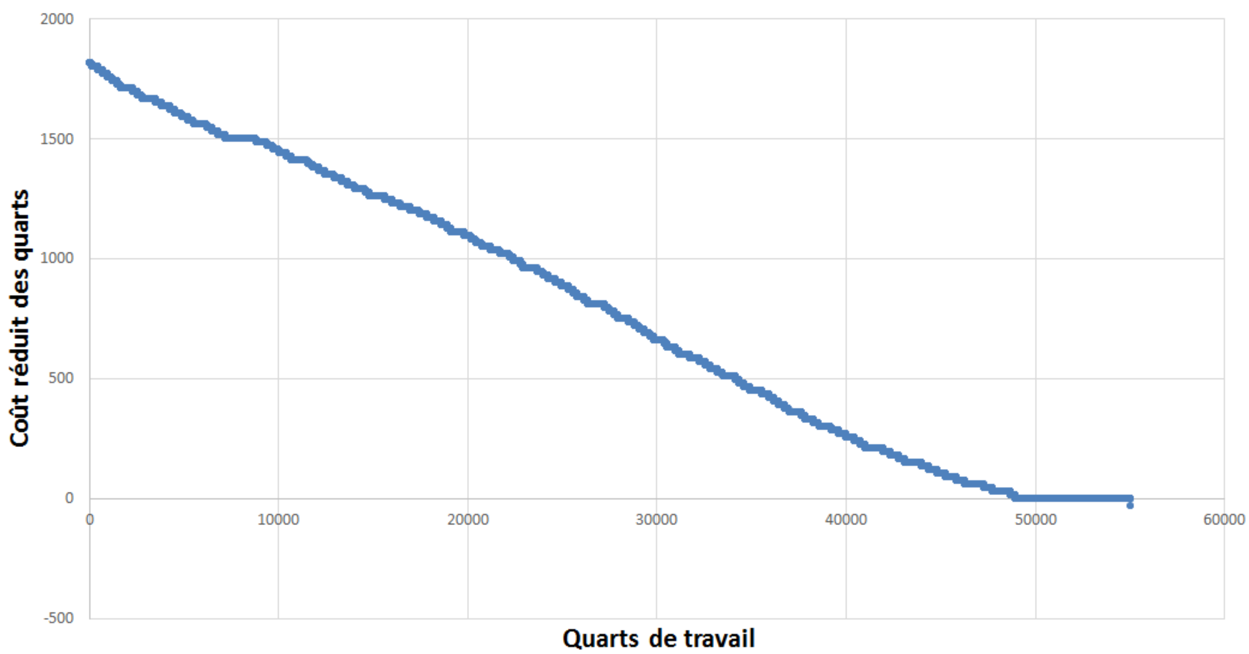

Figure 4.2 Distribution des coûts réduits (25 employés, groupe 3)

4.2.4 Fixations de variables

Nous avons envisagé une dernière façon de réduire les temps de calcul. Le but est de fixer certaines variables pour réduire la taille du modèle après la résolution des premières relaxations linéaires. Tant que l'on peu fixer des variables, on recommence la résolution des premières relaxations linéaires. Lorsqu'aucune variable ne peut être fixée selon un ensemble de paramètres, le solveur résout le problème résiduel avec toutes les variables fixées. Voici les paramètres utilisés :

— un seuil $\sigma \in]0, 1[$ qui correspond à la valeur à partir de laquelle on décide de fixer une variable associée à un quart personnalisé à 1. Les variables sont sélectionnées par ordre décroissant de valeur, et on fixe au plus un quart par employé. Par exemple pour $\sigma = 0.8$, si on a $x_1 = 0.85$, $x_2 = 0.79$, $x_3 = 0.48$, $x_4 = 0.81$ et $x_5 = 0.99$ (chacune étant associée à un quart de différents employés), alors on fixera les variables $x_1 = x_4 = x_5 = 1$;

— $C_{max} \in \mathbb{R}$, qui représente le maximum de la somme des compléments. Le complément est la différence entre 1 et la valeur d'une variable dans $[0, 1]$, donc le maximum de la somme des compléments limite le nombre de variables fixées à 1 grâce au seuil σ. Pour l'exemple ci-haut, si $C_{max} = 0.3$, alors on fixera seulement $x_1 = x_5 = 1$ qui ont des compléments de 0.01 et 0.15. En fixant $x_4 = 1$ qui a un complément de 0.19, C_{max} serait excédé.

Plusieurs tests ont été faits faisant varier σ et C_{max}. Ils peuvent adopter plusieurs valeurs qui devraient avoir des impacts différents. Par exemple $\sigma = 0.6$ fixera un nombre très important de variables comparé à $\sigma = 0.95$. Il devrait donc y avoir des conséquences à la fois sur les temps de calcul et sur les coûts des solutions.

De façon générale, nous avons remarqué que les temps de calcul diminuent, mais moins fortement qu'avec les précédentes stratégies. Pour 25 employés, l'amélioration du temps de calcul est de l'ordre de la seconde, tandis que pour 32 et 47 employés, l'amélioration est de moins de 2%. Rien qu'à cause du côté aléatoire de la mesure des temps de calcul, il est aisé de constater que ces modifications sont négligeables. En effet, nous avons observé qu'il y a généralement peu de variables à fixer à chaque itération. Cela explique que cette stratégie ne réduise que trop peu les temps de calcul. Par conséquent, nous avons décidé de concentrer les résultats sur les autres stratégies proposées. Notons qu'en couplant cette dernière stratégie avec les précédentes, nous n'améliorons pas les temps de calcul, en dégradant parfois les coûts des solutions.

4.3 Résultats des nouvelles approches

Cette section aborde les résultats relatifs aux deux nouvelles approches. Afin d'être complètement exhaustif, nous comparerons les temps de calcul, la satisfaction des préférences et les coûts des solutions avec les résultats du chapitre précédent. Aussi, nous chercherons à montrer l'impact des méthodes visant à diminuer le temps de calcul. À l'issue de cette partie, nous pourrons désigner la meilleure des trois méthodes utilisées. Tous les tests ont été effectués dans les mêmes conditions que ceux de l'approche initiale. On utilise pour la comparaison des groupes de 5 employés.

4.3.1 Approche 1 : ajout de contraintes strictes

Comme cela a été énoncé précédemment, cette approche peut avoir des inconvénients assez importants. Le tableau 4.2 expose les résultats que nous obtenons pour les différentes instances et le tableau 4.1 rappelle les résultats de l'approche initiale. Entre parenthèses, on note l'augmentation en % de la valeur de la fonction objectif par rapport au résultat sans préférence.

Tableau 4.1 Résultats pour l'approche initiale en utilisant des groupes de 5 employés

Instance	25 employés	32 employés	47 employés
Temps (en s)	1203	129	350
Ratio moyen	99.1%	93.1%	96.9%
Coût	345 000 (+121%)	755 430 (+337%)	1 857 864 (+370%)

Tableau 4.2 Résultats pour l'approche 1 en utilisant des groupes de 5 employés

Instance	25 employés	32 employés	47 employés
Temps (en s)	Trop long	367	13761
Ratio moyen	ND	94.2%	97.6%
Coût	ND	554 708 (+220%)	626 672 (+58%)

La première chose que nous pouvons remarquer est l'absence de résultats pour l'instance avec 25 employés. Cela est dû au fait que les temps de calcul sont trop longs pour prendre la peine de chercher un résultat exploitable. L'instance à 32 employés fonctionne plutôt bien en termes de temps de calcul, mais nous noterons quand même que le temps de calcul pour l'approche initiale était de 129 secondes. On a donc multiplié le temps de calcul par 2.85. Pour l'instance de 47 employés, l'optimisation s'est terminée au bout de 3 heures et 49 minutes.

Évidemment ceci n'est absolument pas acceptable comparé à l'approche initiale, même en dépit du fait que les coûts ont été améliorés, tout comme la satisfaction moyenne des employés. Pour ce qui est des plus seniors les 15 premiers sont satisfaits à 100% (voir annexe A). Pour l'approche initiale et 32 employés, c'était le même résultat. Cela signifie que nous avons amélioré la satisfaction moyenne des employés, tout en maintenant le nombre de seniors satisfaits à 100%.

4.3.2 Approche 2 : relaxation des contraintes d'intégrité

De même que pour l'approche précédente, on peut résumer dans un tableau les résultats obtenus lorsque nous relaxons la contrainte d'intégrité sur les employés juniors.

Tableau 4.3 Résultats pour l'approche 1 en utilisant des groupes de 5 employés

Instance	25 employés	32 employés	47 employés
Temps (en s)	740	177	493
Ratio moyen	99.6%	99.9%	99.2%
Coût	159000 (+1.9%)	174898 (+1.1%)	404592 (+2.3%)

Ici les résultats sont impressionnants. Avant tout nous aimerions souligner le coût des solutions. On arrive à augmenter le coût des solutions d'au pire 2.3% par rapport aux solutions sans préférence. Cela correspond au respect d'une contrainte importante du cahier des charges. De plus, même si nous allongeons le temps de calcul, ceux-ci restent relativement raisonnables, comme le soulignent les 493 secondes utilisées pour résoudre l'instance à 47 employés.

Enfin, remarquons que la satisfaction des préférences est largement meilleure que pour n'importe quelle approche utilisée auparavant. Cela est largement dû au fait que la distribution des activités se fait de bien meilleure manière. Dans le cas précis de l'instance avec 32 employés, les 11 employés les plus seniors sont satisfaits à 100%. Ce résultat est moins bon que pour les autres approches (les 15 premiers) mais les deux seuls employés qui ne sont pas satisfaits à 100% le sont à 98% et 99.2%. Pour les personnes en charge des horaires, il est tout à fait concevable d'établir un arbitrage entre le nombre d'employés parmi les plus seniors à être satisfaits à 100% et la satisfaction moyenne.

4.3.3 Impact des méthodes pour réduire les temps de calcul

Dans cette sous-partie, nous essayons de mesurer l'impact des coupes, de la relaxation de la contrainte de repos minimum, et du retrait des quarts ayant un coût réduit trop élevé sur le temps de calcul. Il convient de rappeler que l'ajout de coupes pour réduire le nombre de quarts du problème concerne uniquement l'approche 1, et la relaxation de la contrainte de repos concerne uniquement l'approche 2. En aucun cas, ces deux méthodes peuvent être combinées. Dans le premier cas, il n'y a pas de contrainte à relâcher, car nous utilisons des quarts anonymes pour combler la demande. Et dans le second, il est absolument inutile d'utiliser les coupes car les horaires des seniors sont fixés à chaque itération. Ainsi, nous distinguons bien l'impact des coupes et celui de la relaxation des contraintes de repos minimum. En revanche, la réduction du nombre de quarts par le retrait des quarts dont le coût réduit est supérieur à un paramètre choisi est applicable aux deux approches. Dans tout ce qui suit, nous allons utiliser les abréviations suivantes :

— TaQ signifie Temps de calcul avec retrait de Quarts ;
— TaR signifie Temps de calcul avec Relaxation de la contrainte de repos minimum ;
— TQR signifie Temps de calcul avec retrait de Quarts et Relaxation de la contrainte de repos minimum.

Coupes Cette méthode peut s'avérer assez efficace, malheureusement nous n'avons pas assez de matière pour la tester. En effet, nous n'avons pas pu la tester avec 25 employés dans la mesure où nous n'avons pas de point de repère.

Pour l'instance à 47 employés, les temps de calcul annoncés précédemment sont ceux avec les coupes. Par conséquent, les 3 heures et 41 minutes de temps de calcul doivent être confrontés aux 6 heures et 41 minutes de temps de calcul sans les coupes. Cela implique que nous avons réussi à réduire de 43% le temps de calcul. Pour 32 employés, remarquons que le résultat en temps de calcul sans les coupes est de 905 secondes. En conséquence, nous avons réussi à diviser le temps de calcul par 2.47, en comparaison avec les 367 secondes du tableau 4.2.

Relaxation de la contrainte de repos minimum Le tableau 4.4 présente les différences entre les temps de calcul avec et sans la relaxation de contraintes de repos minimum.

Notons qu'il y a une amélioration des temps de calcul. En effet, sur l'instance à 25 employés, nous avons divisé le temps par 1,12. Pour celle à 32 employés, nous divisons le temps par 1,17. Pour la dernière instance, nous avons divisé le temps par 1,13. Ces améliorations ont l'air faibles en apparence mais excèdent à chaque fois les 10% ce qui n'est pas négligeable.

Tableau 4.4 Temps de calcul en secondes avec et sans relaxation de la contrainte de repos minimum

Instance	25 employés	32 employés	47 employés
TsR	832	208	560
TaR	740 (-11%)	177 (-15%)	493 (-12%)

De plus, nous n'altérons jamais les coûts des solutions.

Travail sur les coûts réduits À ce stade de l'étude, on peut regarder deux impacts différents du retrait des quarts avec coût réduit trop élevé. D'abord, on peut regarder l'impact de cette méthode sur les temps d'origine, puis l'impact sur les temps obtenus avec les réductions énoncés dans les deux paragraphes précédents. Premièrement, remarquons dans le tableau 4.5 que l'impact est faible sur les solutions déjà retravaillées. Nous avons réalisé plusieurs tests en faisant varier le paramètre v entre 50 et 1. Les variations sur les temps de calcul ne sont pas significatives, même si le temps de calcul tend à décroître légèrement en même temps que v. Dans le tableau 4.5, les tests ont été faits avec $v = 1$.

Tableau 4.5 Comparaison des résultats selon les stratégies de réduction du temps de calcul en secondes et pour $v = 1$ et des coûts associés

Instance	25 employés	32 employés	47 employés
Temps d'origine	832	208	560
TaR	740 (-11%)	177 (-15%)	493 (-12%)
TaQ	735 (-11.6%)	172 (-17.3%)	471 (-16%)
TQR	734 (-11.7%)	172 (-17.3%)	453 (-19%)
Coût TaR	156 000	174 898	404 592
Coût TaQ	159 000	175 183	405 434
Coût TQR	159 000	175 243	417 102

Ce dernier tableau nous renseigne sur plusieurs points. Premièrement, les deux méthodes ont plus ou moins le même effet. Même si la réduction du nombre de quarts a l'air de faire gagner un peu plus de temps, cela reste minime. Deuxièmement, on constate que la combinaison des deux méthodes n'apporte pas beaucoup de gains supplémentaires, excepté pour l'instance de 47 employés. Peut-être alors qu'il est préférable de combiner ces méthodes sur les instances de plus grande taille, mais que cela s'avère inutile pour les instances de petite taille (avec moins de quarts possibles), qui prennent souvent moins de temps à résoudre par ailleurs.

La qualité des solutions (en termes de coût et de satisfaction) demeure bonne quelle que soit la stratégie utilisée. Il y a moins de 3% d'augmentation entre le coût de la solution combinant les deux stratégies de réduction des temps de calcul et le coût de la solution utilisant l'une ou l'autre stratégie. Et ceci est vrai dans le pire des cas (47 employés). Dans le cas de 25 employés, l'écart est de 1.9% et dans celui de 32 employés, l'écart est 0.1%. Par conséquent il n'est pas préjudiciable de combiner ces deux stratégies, néanmoins il n'y a pas grand chose à gagner en temps de calcul. Et ce qui est gagné en temps de calcul a des chances d'être perdu en qualité de la solution.

CHAPITRE 5 ANALYSE DE SENSIBILITÉ

Nos approches de résolution sont heuristiques et reposent sur de nombreux paramètres. Il y a à la fois les coefficients de seniorité (λ_e), mais aussi la taille des groupes qui peuvent être tout à fait variables selon la taille de l'instance à résoudre. Un autre paramètre peut être la tolérance à l'optimalité.

Étant donné que nous avons établi que l'approche 2, qui relaxe les contraintes d'intégrité sur les employés juniors, était la meilleure approche que nous avons proposée, nous décidons d'effectuer une analyse de sensibilité sur celle-ci. Ainsi, nous allons pouvoir montrer l'impact des différents paramètres sur les solutions obtenues. Nous allons également pouvoir montrer les arbitrages que devront faire les opérationnels entre le coût, le temps de calcul et la satisfaction de la préférence choisie. Dans la suite, tous les tests ont été effectués dans les conditions de test suivantes. Nous prenons en compte la relaxation de la contrainte de repos minimum qui réduit les temps de calcul, mais nous n'enlevons pas de quarts au problème.

5.1 Paramètres variables

Commençons par présenter la variabilité de la taille des groupes. Les instances utilisées comportent entre 25 et 47 employés, donc nous avons décidé d'utiliser six tailles de groupes différentes. Ainsi, nous pourrons résoudre nos instances en utilisant des groupes de 3, 4, 5, 6, 7 ou 8 employés. Cela revient à dire que nous donnons plus d'importance à l'ancienneté lorsque nous utilisons des groupes petits, et un peu moins lorsque nous utilisons des grands groupes. Cela est principalement dû au fait que lorsque nous nous intéressons à un groupe petit, il y a de plus fortes chances que l'on donne en priorité des heures aux employés de ce groupe plutôt qu'à ceux du groupe suivant. Au contraire, pour un grand groupe, en fonction des coefficients de seniorité, il est possible que la répartition des heures au sein d'un même groupe soit plus équilibrée, ce qui pourrait défavoriser légèrement les plus seniors, en comparaison avec une résolution par petits groupes. Mais tout ceci est relativement approximatif dans la mesure où nos approches de résolution sont heuristiques, ce qui donne de l'incertitude par rapport aux résultats attendus.

De façon beaucoup plus précise, la taille des groupes risque d'avoir une influence non négligeable sur le temps de calcul des solutions. En effet, pour une instance, il est totalement différent de résoudre 6 fois le problème ou 16 fois (47 employés par groupe de 3 ou de 8 respectivement). Même si les problèmes avec des petits groupes devraient être plus faciles à résoudre pour l'optimiseur, il demeure que cela restera sans doute plus long de traiter le problème 16 fois plutôt que 6.

Parlons à présent des coefficients de seniorité. Ce sont des paramètres clés du projet qui peuvent avoir un très grand impact, à la fois sur la satisfaction de la préférence bien sûr, mais aussi sur les coûts des solutions. En effet, ces derniers tronquent le résultat obtenus par l'optimiseur, en ajoutant des bonus qui font artificiellement décroître la valeur de la fonction objectif. Ainsi, la valeur "réelle" de la fonction objectif peut être très élevée, bien au-delà de la valeur sans préférence. Les résultats montreront qu'il est préférable de bien calibrer les bonus pour éviter des erreurs de jugement grossières. Le tableau 5.1 indique les différents coefficients utilisés pour les instances en fonction de la taille des groupes choisis.

Tableau 5.1 Variation des poids de seniorité pour l'analyse de sensibilité

Variante	Poids maximum	Pas	Poids minimum
Normal	$Taille_{groupe}$	-1	1
div2	$\frac{Taille_{groupe}}{2}$	-0.5	0.5
div10	$\frac{Taille_{groupe}}{10}$	-0.1	0.1
fois2	$Taille_{groupe} * 2$	-2	2
fois10	$Taille_{groupe} * 10$	-10	10

Par exemple, pour l'instance de 25 employés, si nous considérons des groupes de 5 employés et la variante *Normal*, les coefficients seront les suivants :
- 5, 4, 3, 2, 1 pour les employés du premier groupe, du plus senior au moins senior ;
- Les mêmes coefficients pour les deuxième, troisième et quatrième groupe ;
- 5, 4, 3, 2, 1 pour les employés du cinquième groupe, du plus senior au moins senior.

Si nous avions choisi des groupes de 7 employés et la variante *fois2*, les coefficients seraient :
- 14, 12, 10, 8, 6, 4, 2 pour les employés du premier groupe, du plus senior au moins senior ;
- Les mêmes coefficients pour les deuxième et troisième groupes ;
- 0.04, 0.03, 0.02, 0.01 pour les employés du cinquième groupe, du plus senior au moins senior.

Le choix de ces coefficients dans un dernier groupe de plus petite taille est fait de sorte à être sûr de ne pas avantager ces derniers employés, qui sont les plus juniors. Nous procédons comme cela pour le dernier groupe, à chaque fois qu'il est plus petit que les autres. Dans le cas d'un dernier groupe de taille plus petite que les autres, nous choisirons les coefficients de ce dernier groupe de la manière suivante : le plus senior aura un coeffient égal à $\frac{Taille_{derniergroupe}}{100}$ et le pas entre chaque employé sera égal à 0.01 de sorte à ce que le dernier employé ait un coefficient de 0.01.

5.2 Résultats

Dans cette section, nous abordons les résultats obtenus pour cette analyse de sensibilité. Nous analysons par la suite l'impact des différents paramètres sur les solutions et terminons par discuter de l'arbitrage à choisir entre les trois objectifs du cahier des charges : coût, temps de calcul et satisfaction des employés pa rapport au nombre d'heures travaillées.

5.2.1 Résultats généraux

Les tableaux 5.2, 5.3 et 5.4 répertorient tous les tests effectués pour mener cette analyse de sensibilité. Dans les paragraphes suivants, nous allons analyser ces résultats et tirer les conclusions possibles. Nous avons relevé les informations suivantes : la variante des poids de seniorité évoquée dans le tableau 5.1, la taille des groupes utilisée, le temps de calcul, le coût de la solution, la satisfaction moyenne et le rang de seniorité du premier employé qui n'est pas satisfait à 100%. Lorsqu'il y a la mention ND, cela signifie que tous les employés ont un ratio égal à 100%

Tableau 5.2 Résultats pour 25 employés

Variante	*Taille des groupes*	*Temps (en s)*	Coût	*Satisfaction moyenne*	*Premier insatisfait*
Normal	3	1138	156000	99,6	7
	4	1141	156000	99,6	7
	5	708	156000	99,6	3
	6	812	162389	96,3	3
	7	949	162389	98,8	2
	8	643	166270	96,4	7
div2	3	1691	156000	97,7	1
	4	1130	156000	99	1
	5	1082	156000	99,4	3
	6	1005	156000	96,64	6
	7	699	156000	94,44	1
	8	641	156000	96,12	3
div10	3	1399	156000	99,44	2
	4	1317	157500	98,6	2
	5	867	156000	99,5	4
	6	945	156000	98,12	5
	7	636	156000	98	3
	8	652	156000	96,28	2
fois2	3	1433	162389	98,28	1
	4	1235	176770	98	3
	5	714	179770	99,9	6
	6	780	191770	96,76	6
	7	874	190270	99,44	14
	8	851	202270	97,64	8
fois10	3	1624	230570	100	ND
	4	1177	257960	97,04	6
	5	937	304909	99,8	5
	6	936	270073	99,5	18
	7	718	259130	99,3	14
	8	722	287240	99,8	5

Tableau 5.3 Résultats pour 32 employés

Variante	*Taille des groupes*	*Temps (en s)*	Coût	*Satisfaction moyenne*	*Premier insatisfait*
Normal	3	282	173813	99,9	19
	4	203	175358	99,7	15
	5	177	174898	99,9	12
	6	156	179162	99,7	15
	7	138	175408	99,9	7
	8	102	180523	99,7	15
div2	3	279	172973	99,9	19
	4	200	172913	99,7	15
	5	186	173743	99,9	17
	6	270	174343	98,9	7
	7	258	174032	98,5	8
	8	109	175943	99,7	15
div10	3	282	172913	99,9	19
	4	212	172913	99,9	19
	5	182	172913	99,3	7
	6	169	172993	99,5	7
	7	135	172913	99,6	8
	8	112	172913	99,6	15
fois2	3	271	186006	100	ND
	4	218	182188	99,7	15
	5	183	180077	99,65	19
	6	270	190937	98,7	8
	7	124	193348	97,6	8
	8	99	212424	100	ND
fois10	3	249	265916	100	ND
	4	181	284694	100	ND
	5	168	251021	99,7	25
	6	140	293978	100	ND
	7	127	315124	98,8	28
	8	100	348890	100	ND

Tableau 5.4 Résultats pour 47 employés

Variante	*Taille des groupes*	*Temps (en s)*	Coût	*Satisfaction moyenne*	*Premier insatisfait*
Normal	3	769	402815	99,9	11
	4	566	405038	99,85	4
	5	493	404592	99,2	4
	6	373	410981	99,9	4
	7	332	444213	99,4	18
	8	296	436717	99,9	4
div2	3	785	409196	99,1	2
	4	543	426854	99,8	2
	5	502	413647	99,4	18
	6	416	412912	99	18
	7	340	406165	99,9	18
	8	309	425849	99,8	4
div10	3	803	408956	99,55	4
	4	592	421814	99,85	18
	5	542	408626	99,3	18
	6	416	405716	99,5	8
	7	389	408896	100	ND
	8	319	449054	99,5	11
fois2	3	767	425203	99,87	19
	4	567	445320	99,4	4
	5	487	418561	99,2	4
	6	353	428974	100	ND
	7	331	430290	100	ND
	8	294	442458	100	ND
fois10	3	786	490082	100	ND
	4	554	511919	100	ND
	5	544	570950	100	ND
	6	416	564841	99,9	11
	7	380	596409	100	ND
	8	289	744261	99,9	42

Impact des coefficients de seniorité En premier lieu, intéressons-nous à l'impact des coefficients de seniorité sur les solutions. À la lumière des tableaux précédents, il est aisé de remarquer deux choses. La première est que les temps de calcul sont assez similaires en fonction de la variante utilisée. Les différences entre deux variantes pour une même taille de groupe est trop faible pour indiquer un quelconque impact des coefficients sur cet aspect. La deuxième est que l'impact sur les coûts des solutions ainsi que sur la satisfaction moyenne est bien réel.

Pour ce qui est du coût des solutions, le constat semble clair. Plus les coefficients sont élevés(cf *fois2* et *fois10*), plus les coûts des solutions sont grands. Pour rappel, les coûts des solutions ne prennent pas en compte les bonus, qui ne constituent que des coûts purement fictifs. Ce constat est criant pour le jeux de données comprenant 25 employés, mais remarquons que cela se vérifie pour les autres instances comme l'indique la figure 5.1 :

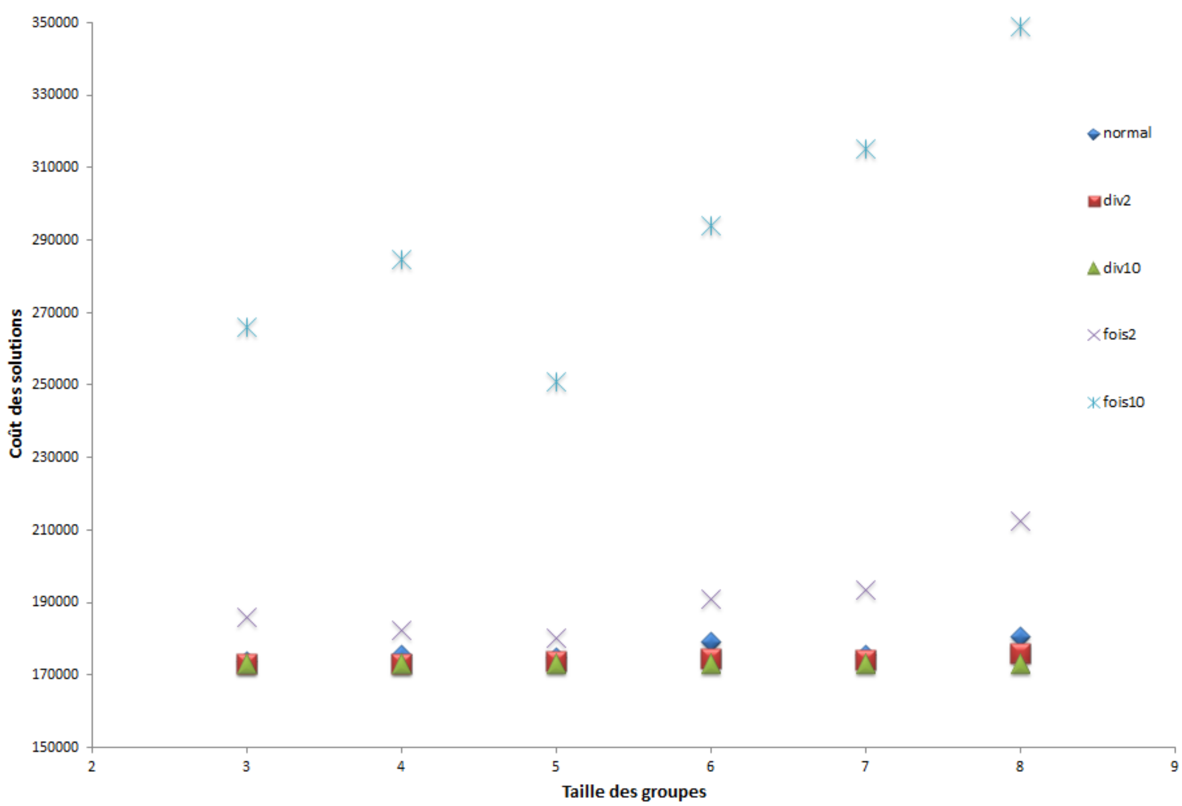

Figure 5.1 Impact des coefficients de seniorité sur le coût des solutions pour 32 employés

Clairement, les variantes *fois2* et *fois10* induisent des coûts qui ne sont pas appréciables. Ce sont les bonus trop élevés qui viennent compenser la sur-couverture et la sous-couverture.

Ainsi, le résultat est faussé, et c'est pourquoi il faut utiliser les bonus avec précaution. En tout cas, les trois autres variantes apportent des résultats acceptables, le plus souvent moins de 2% plus élevés que l'optimum.

À l'inverse, la satisfaction des préférences est moins importante lorsque les coefficients sont trop bas. Si nous prenons en exemple l'instance de 47 employés, on peut remarquer que la satisfaction moyenne est meilleure pour les variantes *fois2* et *fois10*. En effet, pour ces variantes, il n'est pas rare d'obtenir un respect des préférences de 100%. Aussi, le rang du premier employé non satisfait à 100% est plus élevé en général. Ce constat s'applique également aux deux autres instances. Néanmoins, il convient de remarquer que les préférences sont globalement très bien satisfaites, et ce quelle que soit l'instance utilisée. En effet, la pire moyenne enregistrée est 96.12% pour l'instance à 25 employés, 97.6% pour 32 et 99.2% pour 47. De plus, même lorsque ce sont les plus seniors qui sont lésés, leur satisfaction ne descend jamais en dessous de 95%. Pour rappel, cela signifie qu'ils travaillent 95% du maximum d'heures possible. Si ce maximum est de 40 heures, alors l'employé en question travaille 38 heures, et c'est le pire cas possible. Les rares fois où une satisfaction est basse (i.e. inférieure à 90%), cela concerne un employé relativement junior (i.e. pas dans les 50% les plus seniors).

Ainsi, bien que l'impact soit faible au regard de la satisfaction des préférences, à cause du taux de satisfaction très élevé partout, il est quand même non négligeable. Au regard de tout ce qui a été énoncé, un gain faible sur la satisfaction des préférences peut en revanche engendrer un coût des solutions bien plus importants, ce qui peut s'avérer rédhibitoire en contexte opérationnel.

Un autre moyen de comparer les différentes variantes proposées consiste à établir des profils de performance. Nous proposons un de ces graphiques inspiré des travaux de Dolan et Moré (2002) ainsi que ceux de Moré et Wild (2009). L'idée est de représenter la proportion de solutions acceptables selon une définition établie en fonction du coût des solutions. Nous décidons de normaliser les coûts des solutions entre 0 et 100. Ainsi, pour chaque instance, le 0 en abscisse (paramètre α) correspond au coût minimum que nous avons obtenu, et 100 correspond au coût maximum. Par conséquent, un point qui a pour abscisse 28, est à 28% entre le coût minimum et le coût maximum pour l'instance considérée.

Il faut que nous nous donnions une définition de solution acceptable. Les temps de calcul étant relativement équivalents selon les variantes, il n'est pas nécessaire de les inclure dans la définition. On appellera solution acceptable, une solution qui vérifie les deux critères suivants :

— Le ratio moyen de satisfaction doit être supérieur ou égal à 99%

— Le rang du premier employé insatisfait doit être strictement supérieur à $\frac{nb_{employés}}{10}$

Ainsi, sur les 18 solutions obtenues au cours des tests pour chaque variante (6 tailles de groupes différentes et 3 instances différentes), il est possible de déterminer la proportion de solutions acceptables, et pour quel coût. Le figure 5.2 traduit les résultats des tableaux des pages précédentes, selon la définition de solution acceptable donnée.

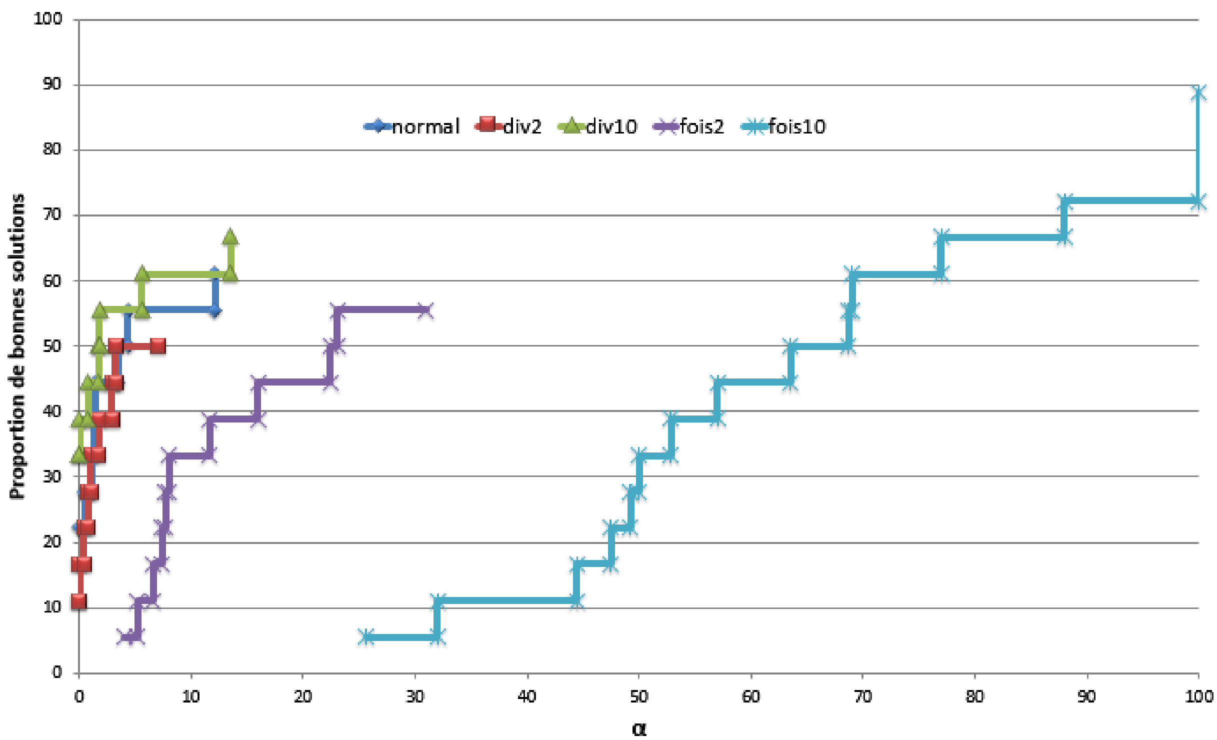

Figure 5.2 Profils de performance pour chaque variante

L'analyse de ce graphique réside en trois différents points :

— Est-ce qu'une courbe domine les autres ?

— Que se passe-t-il en 0, c'est-à-dire à coût minimum ?

— Que se passe-t-il en 100, c'est-à-dire à coût maximum ?

Ces trois points nous donnent toute l'information nécessaire pour résumer les trois tableaux de résultats présentés auparavant. Au regard du profil de performance proposé, nous pouvons répondre à la première question en affirmant que la courbe verte domine les autres. En, effet elle est au dessus de toutes les autres sauf à la limite à droite. A priori, cela nous indique que c'est la meilleure variante proposée. De plus, si on observe ce qu'il se passe en 0, on remarque que c'est également celle qui produit le plus de bonnes solutions à coût minimum. Les variantes *fois2* et *fois10* ne proposent aucune solution acceptable à coût minimum. Donc il y a des chances que ce soit les plus mauvaises. Notons qu'à coût maximum, la variante *fois10* propose 90% de solutions acceptables, ce qui est excellent. Néanmoins ce coût maximum est trop élevé et dépasse largement les coûts voulus à l'origine. Donc on peut affirmer que les variantes imposant des coefficients trop élevés sont à bannir. De plus, les coefficients les plus faibles ont l'air d'être les meilleurs avec les tests effectués.

Impact de la taille des groupes Le second paramètre qui influe majoritairement sur les solutions est la taille des groupes. En effet, pour l'opérateur, il est possible de choisir le nombre d'employés à traiter en même temps. Plus les groupes sont de petite taille, plus il y aura d'itérations au problème. Et cela impacte directement les temps de calcul. La figure 5.3 présente la relation entre la taille des groupes et le temps de calcul d'une part, le coût des solutions d'autre part.

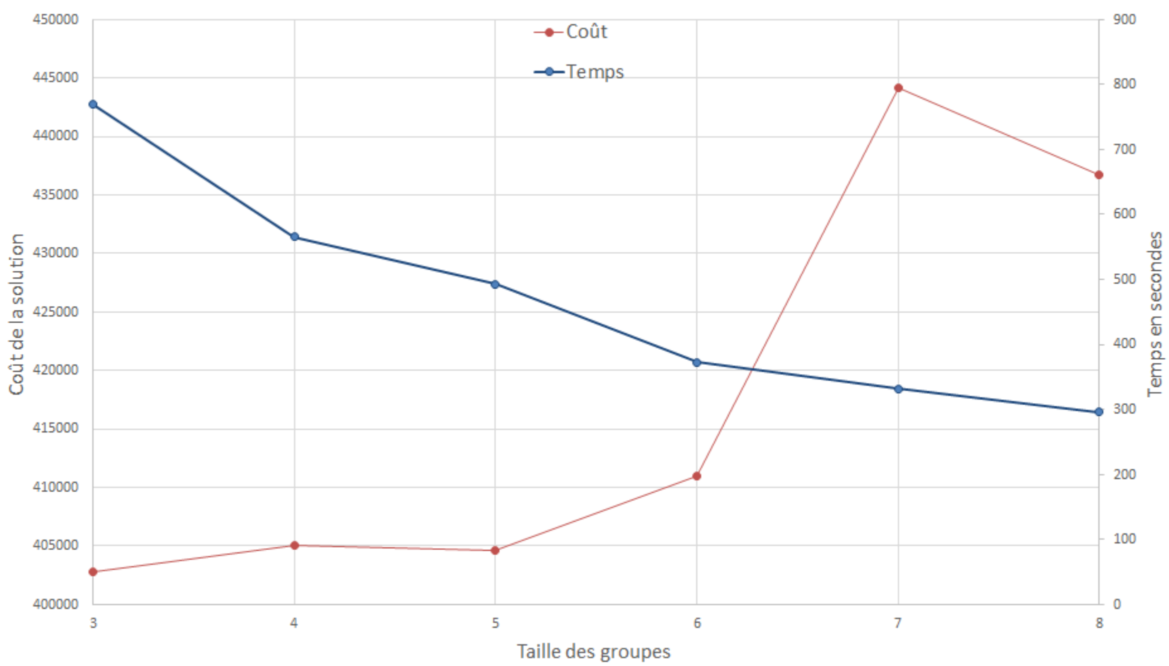

Figure 5.3 Impact de la taille des groupes sur le coût des solutions et le temps de calcul pour 47 employés

La relation entre la taille des groupes est évidente à discerner. On constate que plus les groupes sont petits, plus les temps de calcul sont longs. Dans le cas de la figure 5.3, on divise par 2.5 le temps de calcul entre des groupes de 3 et de 8 employés. Ce constat est très important, surtout lorsque les instances vont s'agrandir, car on pourra diviser le nombre de problème résolus de façon importante, et ainsi faire de gros gains en temps de calcul.

En revanche, en observant l'évolution des coûts, nous pouvons constater qu'ils ont tendance à augmenter avec la taille des groupes. Premièrement, ceci n'est pas toujours vrai. Pour d'autres tests (par exemple 47 employés, variante *fois2*), le coût le plus élevé peut être engendré en utilisant des groupes de 4 employés. Cette variabilité au niveau du coût est principalement due à deux choses. D'une part, lorsque la taille des groupes est grande, la moyenne des bonus est plus élevée. En effet, les bonus seront plus élevés en faisant trois groupes avec 8, 7, 6, 5, 4, 3, 2, 1 plutôt qu'avec 8 groupes dont les coefficients sont 3, 2, 1. Ceci peut expliquer en partie le fait que les coûts sont plus élevés parfois pour les groupes de grande taille. D'autre part, les préférences ont des chances d'être mieux respectées en faisant des petits groupes. Il sera plus facile de donner le maximum d'heures à 3 employés, alors qu'avec 8, il y a plus de chances d'équilibrer le nombre d'heures, au détriment des plus seniors. Cette satisfaction améliorée peut engendrer des coûts dans la solution qui expliquent que parfois, les coûts sont plus élevés pour les petits groupes.

Finalement, ces deux effets contradictoires ne nous permettent pas vraiment de conclure quant à l'impact de la taille des groupes sur les coûts. Par contre, nous sommes certains de l'impact sur les temps de calcul, ce qui nous laisse penser que nous pourrions éliminer les tailles de groupes extrêmes. En effet, des groupes de 3 employés seraient trop longs à calculer, et des groupes de 7 ou 8 employés auraient des chances de donner des coûts trop élevés. Pour pallier ce problème, une solution serait de normaliser les coefficients de seniorité pour obtenir une moyenne égale entre chaque taille de groupe. Le souci que cela causerait, serait une détérioration de la satisfaction des préférences notamment au sein d'un même groupe, où les différences de poids entre le premier et le dernier employé doivent être non négligeables pour favoriser les seniors.

5.3 Quel arbitrage choisir ?

Dans cette section, nous récapitulons les variations entre les solutions. Nous allons exposer dans un graphique la position de chaque solution obtenue vis-à-vis de la satisfaction, le coût et le temps de calcul.

Sur la figure 5.4, nous observons pour chaque test de l'instance à 47 employés, le coût et le temps de calcul. En se référant à la solution sans préférence évoquée au chapitre 3, et si l'on

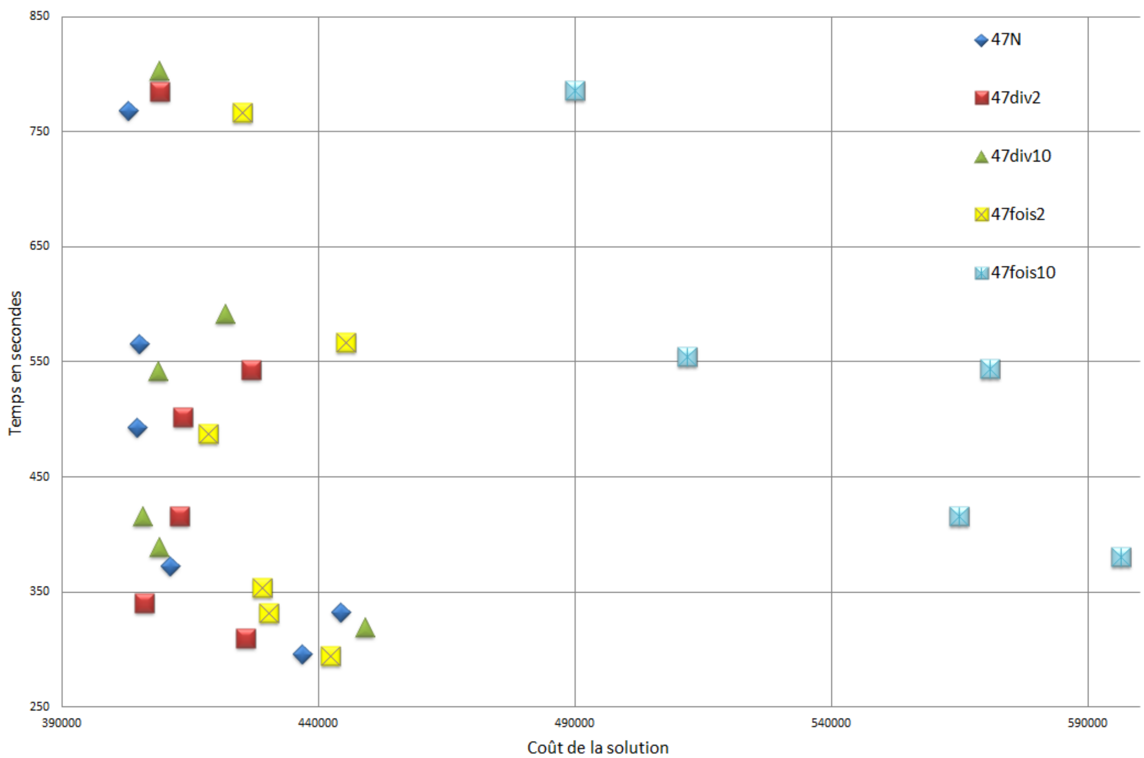

Figure 5.4 Temps de calcul en fonction du coût pour chaque solution de l'instance à 47 employés

s'autorise à dégrader la solution de 5%, il nous faut obtenir un coût inférieur à 415000. Il y a 12 solutions sur 30 qui satisfont ce critère. Parmi elles, aucune pour les variantes *fois2* et *fois10*, ce qui confirme qu'elles sont trop extrêmes pour le cadre de l'étude. Il en reste 4 sur 6 pour chacune des autres variantes. Dans chacun des cas, le coût est trop élevé lorsque nous résolvons par groupes de 8 employés, ce qui confirme la conclusion énoncée à la section précédente.

Sur les 12 solution restantes, il y en a 3 pour lesquelles le temps de calcul est supérieur à 750 secondes (12m30). Idéalement, et dans le cadre du cahier des charges, nous recherchons des temps de calcul inférieurs à 10 minutes. Ainsi 9 solutions pourraient satisfaire le cahier des charges. Les voici par coût décroissant :

- *div2*, groupes de 5 employés
- *div2*, groupes de 6 employés
- *normal*, groupes de 6 employés
- *div10*, groupes de 7 employés
- *div10*, groupes de 5 employés

- *div2*, groupes de 7 employés
- *div10*, groupes de 6 employés
- *normal*, groupes de 4 employés
- *normal*, groupes de 5 employés

Ceci confirme qu'il est mieux d'utiliser des groupes de 5 ou 6 employés, même s'il est possible que des groupes de 7 ou de 4 puissent donner de bons résultats. Aussi, pour les coefficients, il n'y a pas spécialement de préférence à donner entre les trois variantes. A noter que les meilleures solutions sont celles le plus en bas et le plus à gauche dans le graphique. Ainsi, pour chacune des trois variantes, nous obtenons une très bonne solution à la fois en termes de temps de calcul, et de coût de la solution.

Parmi ces trois solutions, on peut essayer de comparer la satisfaction moyenne ainsi que le rang du premier employé non satisfait à 100%. La satisfaction moyenne correspondant au point bleu (*normal*) est égale à 99.2% et le rang de l'employé est 4. Pour le point vert (*div10*), la satisfaction moyenne est de 99.5% et le rang est 8. Enfin, pour le point rouge (*div2*), la satisfaction est de 99.9% et c'est le 18ème employé qui n'est pas satisfait à 100%. Pour les deux premiers, cela correspond aux satisfactions moyennes parmi les plus mauvaises, même si elles restent tout à fait acceptables. Pour le dernier, la satisfaction est quasiment parfaite, seul, l'employé 18 n'est pas satisfait à 100% (il l'est à 97.3%).

La conclusion globale de cette analyse est qu'il est nécessaire d'effectuer un arbitrage entre les différents critères que nous nous imposons. En effet, même après avoir éliminé de nombreuses solutions, parmi les trois restantes, celle qui a le coût le plus faible est aussi la plus longue à obtenir et celle qui satisfait le moins les préférences. À l'opposé, celle qui a le meilleur temps de calcul et la meilleure satisfaction est celle dont le coût est le plus important. Enfin, la troisième est entre les deux premières en tout point. La satisfaction des préférences et le coût étant relativement aléatoire en fonction des tests effectués, le seul vrai levier d'action est le temps de calcul. Si la volonté est d'être très rapide, il faut utiliser des groupes de taille importante, alors que si le temps de calcul est secondaire, il est possible de prendre des groupes plus petits en espérant augmenter les préférences. Quoi qu'il en soit, il faut bannir les coefficients de seniorité trop élevés, ainsi que les petits groupes, sinon quoi le temps de calcul risque d'augmenter considérablement.

CHAPITRE 6 CONCLUSION

Dans le cadre de cette maîtrise, nous avons proposé des solutions au problème de la confection d'horaires du personnel avec préférences et priorités dans le milieu de la vente au détail. Cette conclusion nous permet de proposer une synthèse des divers travaux qui ont été menés, un regard critique sur les limites de notre étude, ainsi que des pistes d'amélioration quant à une éventuelle suite au projet.

6.1 Synthèse des travaux

Nous avons proposé plusieurs étapes dans notre démarche.

- **Approche séquentielle :** Cette méthode était une heuristique basée sur de la programmation en nombres entiers. On décomposait l'ensemble des employés en sous-groupes pour deux raisons : à la fois, cela facilitait la satisfaction des préférences, mais aussi cela améliorait les temps de calcul. Dans un contexte opérationnel, cela représente un avantage indéniable.
- **Évaluation :** Cette étape ne s'inscrivait pas dans un but opérationnel, donc les temps de calcul associés à cette étape n'importent pas. En effet, elle permettait de valider le résultat de l'approche séquentielle. Elle a pour but d'assurer au client que ses employés les plus seniors ne peuvent se plaindre d'avoir obtenu moins de temps de travail que certains plus juniors. Basée sur le *Preferential Bidding System*, elle a ses limites mais convient bien dans le cadre de l'étude.
- **Nouvelles approches :** Cette étape visait à réduire les coûts des solutions proposées. Effectivement, la première étape induisait des approximations dans son approche du problème. Les coûts engendrés étaient trop élevés. Cette deuxième approche permettait d'améliorer cet aspect sans détériorer la satisfaction des préférences des employés.
- **Analyse de sensibilité :** Elle a permis de tester la robustesse de notre meilleure approche face à la variation de certains paramètres du problème. À défaut d'apporter une conclusion sans équivoque, elle a eu le mérite d'écarter les coefficients de seniorité trop élevés. De plus, elle met en valeur l'arbitrage que devra réaliser le gérant du personnel au moment de construire les horaires des employés.

6.2 Limitations de la solution proposée

Malgré tout le travail apporté au projet, il paraît nécessaire d'évoquer les limites de notre étude. Dans un premier temps, nous pouvons signaler qu'il y a une notion d'arbitrage qui intervient dans l'étude. En effet, il y a un lien de cause à effet évident entre la satisfaction des préférences et le coût de la solution. Par exemple si nous ne nous autorisons qu'à dégrader le coût de 2% par rapport à une solution sans préférence, alors les préférences ne pourront qu'être difficilement satisfaites. À l'inverse, si nous nous interdisons de ne pas satisfaire les préférences des employés à moins de 95%, il est impossible de ne pas altérer de façon assez sensible le coût. Une méthode qui réduit, voire élimine cet arbitrage serait préférable à la nôtre.

Dans un deuxième temps, nous pouvons mettre l'accent sur le fait que notre étude ne traite qu'une seule préférence à la fois. Dans la réalité, il existe de multiples préférences différentes et qui ne se satisfont pas de la même façon. Par conséquent, notre modèle d'origine nécessiterait au moins une refonte partielle pour assouvir les besoins liés à l'ajout d'autres préférences.

Dans un troisième temps, il convient de dire que notre échantillon de test est perfectible. En effet, nous avons travaillé sur seulement trois instances pour lesquelles nous avons fait varier de nombreux paramètres. Travailler sur d'autres instances de la vie réelle pourrait donner encore plus de crédit aux travaux effectués. Bien sûr, travailler sur des instances de plus grande taille pourrait également contribuer à l'efficacité du modèle.

6.3 Améliorations futures

En conséquence de ce qui a été dit précédemment, voici un petit récapitulatif des divers apports qu'il pourrait être intéressant d'intégrer au projet par la suite. Premièrement, il serait peut-être pertinent de chercher une méthode qui offre un meilleur ratio coût/satisfaction des préférences. En effet, à ce jour, il est nécessaire de s'imposer un arbitrage entre les coûts et la gestion des préférences. Pouvoir s'absoudre de cet arbitrage serait une avancée conséquente pour le projet.

Deuxièmement, et c'est sans doute le plus important, il serait très intéressant de pouvoir ajouter d'autres préférences au problème. Au hasard, il faudrait ajouter les préférences concernant les jours de repos/travail, la volonté de faire des quarts plus longs ou la volonté de faire des quarts plus tôt. Le modèle actuel ne fonctionnant pas pour comparer des heures travaillées, et le fait d'avoir eu ou non les jours de repos demandés, il faudra procéder à une refonte permettant d'intégrer diverses préférences ne se comparant pas à la même échelle. C'est dans ce cadre-là que le travail mené par Bana e Costa et Vansnick (1994) ou Hoang (2014) devra être mis à contribution. La méthode MACBETH de ces premiers consiste à établir une fonction linéaire d'utilité multi-caractéristiques pour l'employé, en le faisant répondre à des questions du type : *" A quel point préférez-vous commencer un quart de travail à 8h par rapport à 9h ? "* Ainsi, MACBETH (*Measuring Attractiveness by a Categorical Based Evaluation Technique*) peut modéliser les préférences de chaque employé sous la forme d'une somme pondérée de fonctions d'utilité partielles, chaque fonction d'utilité partielle correspondant à un type de préférence. Les calculs de fonction d'utilité regroupant toutes les préférences permettra de revenir à un modèle semblable à celui que nous avons utilisé. Le second propose la méthode AMACSN (*Affine Multi-Attribute Correlated Social Normalization*), plus performante. D'abord, elle permet de trouver une normalisation des fonctions partielles d'utilité cohérente, qui autorise la comparaison entre deux employés. Pour ce faire, l'auteur introduit une nouvelle normalisation qui consiste à comparer, pour une préférence, l'utilité partielle de son niveau de préférence avec le niveau de préférence d'un autre employé. Ensuite, la méthode rééquilibre les poids dans la fonction multi-caractéristiques pour pouvoir obtenir la normalisation finale.

Enfin, une possibilité d'amélioration simple serait de tester ce que nous avons fait sur un plus grand échantillon de données. En somme, il faudrait être capable de confirmer les conclusions que nous avons données sur d'autres instances du même type que les nôtres. Aussi, voir si le modèle s'adapte à des instances de grande taille (100 employés et plus) serait un plus à long terme. Pour l'instant, il faut être conscient que nous n'avons pas idée de l'impact d'une augmentation significative de la taille du problème au regard de la satisfaction des préférences.

RÉFÉRENCES

H. Achour, M. Gamache, F. Soumis, et G. Desaulniers, “An exact solution approach for the preferential bidding system problem in the airline industry”, *Transportation Science*, vol. 41, no. 3, pp. 354–365, 2007.

T. Aykin, “Optimal shift scheduling with multiple break windows”, *Management Science*, vol. 42, no. 4, pp. 591–602, 1996.

C. A. Bana e Costa et J.-C. Vansnick, “Macbeth—an interactive path towards the construction of cardinal value functions”, *International Transactions in Operational Research*, vol. 1, no. 4, pp. 489–500, 1994.

J. F. Bard et H. W. Purnomo, “Preference scheduling for nurses using column generation”, *European Journal of Operational Research*, vol. 164, no. 2, pp. 510–534, 2005.

S. E. Bechtold et L. W. Jacobs, “Implicit modeling of flexible break assignments in optimal shift scheduling”, *Management Science*, vol. 36, no. 11, pp. 1339–1351, 1990.

K. Boubaker, G. Desaulniers, et I. Elhallaoui, “Bidline scheduling with equity by heuristic dynamic constraint aggregation”, *Transportation Research Part B : Methodological*, vol. 44, no. 1, pp. 50–61, 2010.

M. Bouchard, “Coloration de graphes et attribution d’activités dans des quarts de travail”, Thèse de doctorat, École Polytechnique de Montréal, 2008.

J. Byrne, “A preferential bidding system for technical aircrew”, *AGIFORS Symposium Proceedings*, vol. 28, pp. 87–99, 1988.

M.-C. Côté, B. Gendron, et L.-M. Rousseau, “Modeling the regular constraint with integer programming”, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 29–43, 2007.

G. B. Dantzig, “Letter to the editor—a comment on edie’s “traffic delays at toll booths””, *Journal of the Operations Research Society of America*, vol. 2, no. 3, pp. 339–341, 1954.

P. R. Day et D. M. Ryan, “Flight attendant rostering for short-haul airline operations”, *Operations Research*, vol. 45, no. 5, pp. 649–661, 1997.

S. Demassey, G. Pesant, et L.-M. Rousseau, “Constraint programming based column generation for employee timetabling”, dans *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, R. Barták et M. Milano, éds., 2005.

E. D. Dolan et J. J. Moré, “Benchmarking optimization software with performance profiles”, *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

L. C. Edie, “Traffic delays at toll booths”, *Journal of the Operations Research Society of America*, vol. 2, no. 2, pp. 107–138, 1954.

I. Elhallaoui, A. Metrane, F. Soumis, et G. Desaulniers, “Multi-phase dynamic constraint aggregation for set partitioning type problems”, *Mathematical Programming*, vol. 123, no. 2, pp. 345–370, 2010.

A. T. Ernst, H. Jiang, M. Krishnamoorthy, et D. Sier, “Staff scheduling and rostering : A review of applications, methods and models”, *European Journal of Operational Research*, vol. 153, no. 1, pp. 3–27, 2004.

M. Gamache, F. Soumis, D. Villeneuve, J. Desrosiers, et E. Gelinas, “The preferential bidding system at air canada”, *Transportation Science*, vol. 32, no. 3, pp. 246–255, 1998.

M. Gamache, F. Soumis, G. Marquis, et J. Desrosiers, “A column generation approach for large-scale aircrew rostering problems”, *Operations Research*, vol. 47, no. 2, pp. 247–263, 1999.

W. Glanert, “A" timetable" approach to the assignment of pilots to rotations”, *AGIFORS Symposium Proceedings*, vol. 24, pp. 396–391, 1984.

L. N. Hoang, “Conception bayésienne de mécanismes et quantification de l’équité appliquées à la construction d’horaires personnalisés”, Thèse de doctorat, École Polytechnique de Montréal, 2014.

P. Jeandroz, “Heuristique pour la construction de blocs mensuels personnalisés d’agents de bord”, Mémoire de maîtrise, École Polytechnique de Montréal, 2001.

Q. Lequy, G. Desaulniers, et M. M. Solomon, “A two-stage heuristic for multi-activity and task assignment to work shifts”, *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 831–841, 2012.

H. Michon-Lacaze, “Élaboration de quarts de travail robustes aux perturbations de courtes durées”, Mémoire de maîtrise, École Polytechnique de Montréal, 2015.

H. E. Miller, W. P. Pierskalla, et G. J. Rath, “Nurse scheduling using mathematical programming”, *Operations Research*, vol. 24, no. 5, pp. 857–870, 1976.

R. Moore, J. Evans, et H. Noo, “Computerized tailored blocking”, *AGIFORS Symposium Proceedings*, vol. 18, pp. 343–361, 1978.

J. J. Moré et S. M. Wild, “Benchmarking derivative-free optimization algorithms”, *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009.

Z. Omari, “Attribution des activités aux employés travaillant sur des quarts”, Mémoire de maîtrise, École Polytechnique de Montréal, 2003.

D. M. Ryan, “The solution of massive generalized set partitioning problems in aircrew rostering”, *Journal of the Operational Research Society*, pp. 459–467, 1992.

J. St-Germain, *Horaires personnalisés avec priorité : considération accrue des employés juniors*, Mémoire de maîtrise, École Polytechnique de Montréal, 2004.

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, et L. De Boeck, “Personnel scheduling : A literature review”, *European Journal of Operational Research*, vol. 226, no. 3, pp. 367–385, 2013.

E. Vatri, “Intégration de la génération de quarts de travail et de l’attribution d’activités”, Mémoire de maîtrise, École Polytechnique de Montréal, 2003.

ANNEXE A RÉSUMÉ DES RÉSULTATS SUR LA SATISFACTION DES PRÉFÉRENCES

Tableau A.1 Informations sur les évaluations

Instance	Ratio moyen	Pps 100%	Pps 95%	Nombre d'insatisfaits
25 employés				
Initiale	99,1	17	18	2
Ajout	ND	ND	ND	ND
Relaxation	99,6	3	ND	3
32 employés				
Initiale	93,1	16	16	4
Ajout	94,18	16	16	3
Relaxation	99,9	12	ND	2
47 employés				
Initiale	96,9	21	22	7
Ajout	97,6	14	14	3
Relaxation	99,2	3	19	5

Dans ce tableau, nous présentons le ratio moyen obtenu avec chaque méthode de résolution, et pour chaque instance résolue par groupes de 5 employés. Les colonnes "Pps 100%" et "Pps 95%" indiquent le rang du premier employé non satisfait à 100% et 95% respectivement. Notons qu'avec l'approche initiale, ce rang est relativement élevé, comparé à la méthode relaxant la contrainte d'intégrité des plus juniors. Néanmoins, le ratio moyen est meilleur dans ce dernier cas, et le nombre d'insatisfaits à tendance à être plus faible. Par exemple pour 47 employés, le 3ème employé n'est pas satisfait à 100%, mais le premier insatisfait à moins de 95% est le 19ème. Si on ajoute à cela que le ratio moyen est de 99.2%, on se rend facilement compte que le nombre d'insatisfait à moins de 95% est très faible. Et ceux qui le sont, ont de bonnes chances d'être relativement bien satisfaits, comparé à ceux de l'approche initiale, pour laquelle le ratio moyen est plus faible et les 20 premiers employés satisfaits à 100%.

ANNEXE B DEMANDE DES ACTIVITÉS 14765 ET 14768 ET RÉPARTITION DES ACTIVITÉS AVEC ET SANS UTILISATION DES PRÉFÉRENCES

Dans cette annexe, nous exposons la demande pour deux activités différentes, telles qu'elles sont fournies en fichier d'entrée. La figure suivante expose la demande à une date et sur une fenêtre de temps précise. Par exemple, la demande est nulle entre minuit et 5h le 4/05, alors qu'elle est égale à 1 entre 5h et 23h. La seconde figure proposant un extrait de demande pour l'activité 14768 fontionne de la même manière. La figure B.3 expose quant à elle la façon dont sont réparties les activités entre les employés lors de l'optimisation avec et sans préférence.


```
<WorkloadDetail headcount="0" startAndEnd="4/05/2015 12:00AM - 4/05/2015 5:00AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/05/2015 5:00AM - 4/05/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/05/2015 11:00PM - 4/06/2015 4:30AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/06/2015 4:30AM - 4/06/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/06/2015 11:00PM - 4/07/2015 4:30AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/07/2015 4:30AM - 4/07/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/07/2015 11:00PM - 4/08/2015 4:30AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/08/2015 4:30AM - 4/08/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/08/2015 11:00PM - 4/09/2015 4:30AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/09/2015 4:30AM - 4/09/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/09/2015 11:00PM - 4/10/2015 4:30AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/10/2015 4:30AM - 4/11/2015 12:00AM"/>
<WorkloadDetail headcount="0" startAndEnd="4/11/2015 12:00AM - 4/11/2015 5:00AM"/>
<WorkloadDetail headcount="1" startAndEnd="4/11/2015 5:00AM - 4/12/2015 12:00AM"/>
```


Figure B.1 Demande pour l'activité 14765

Ici, il s'agit bien de comparer les courbes de demande pour les deux activités considérées. L'une est uniforme pour toute la journée, et l'autre est beaucoup plus variable, avec une fréquence de changement quasiment à chaque période. Ce sont des demandes comme celle de l'activité 14765 qui peuvent poser problème dans la répartition des activités. La figureB.3 démontre bien cela. Lorsque nous utilisons l'approche initiale, c'est l'activité 14768 qui est attribuée aux employés seniors alors que lorsque nous optimisons sans préférence, c'est l'activité 14765. La résolution par groupes tend à repousser le souci causé par l'activité 14765, car nous comblons la demande avec des quarts anonymes. En attribuant l'activité 14768, il y a moins de quarts anonymes à créer donc c'est plus simple et moins coûteux pour le solveur. Néanmoins cela induit des problèmes à long terme, car à l'issue de l'optimisation, aucun employé n'a pu récupérer l'activité 14765. En effet, les seuls employés qualifiés figuraient parmi les plus seniors, mais comme on leur a attribué une autre activité, celle-ci reste non attribuée.

```xml
<WorkloadDetail headcount="0" startAndEnd="4/05/2015 12:00AM - 4/05/2015 5:30AM"/>
<WorkloadDetail headcount="2" startAndEnd="4/05/2015 5:30AM - 4/05/2015 6:00AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 6:00AM - 4/05/2015 6:30AM"/>
<WorkloadDetail headcount="3" startAndEnd="4/05/2015 6:30AM - 4/05/2015 7:30AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 7:30AM - 4/05/2015 7:45AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 7:45AM - 4/05/2015 8:00AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 8:00AM - 4/05/2015 8:30AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 8:30AM - 4/05/2015 8:45AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 8:45AM - 4/05/2015 9:00AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 9:00AM - 4/05/2015 9:15AM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 9:15AM - 4/05/2015 9:30AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 9:30AM - 4/05/2015 10:15AM"/>
<WorkloadDetail headcount="7" startAndEnd="4/05/2015 10:15AM - 4/05/2015 11:00AM"/>
<WorkloadDetail headcount="9" startAndEnd="4/05/2015 11:00AM - 4/05/2015 11:15AM"/>
<WorkloadDetail headcount="8" startAndEnd="4/05/2015 11:15AM - 4/05/2015 11:30AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 11:30AM - 4/05/2015 11:45AM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 11:45AM - 4/05/2015 12:00PM"/>
<WorkloadDetail headcount="7" startAndEnd="4/05/2015 12:00PM - 4/05/2015 12:30PM"/>
<WorkloadDetail headcount="8" startAndEnd="4/05/2015 12:30PM - 4/05/2015 12:45PM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 12:45PM - 4/05/2015 1:00PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 1:00PM - 4/05/2015 1:15PM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 1:15PM - 4/05/2015 1:30PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 1:30PM - 4/05/2015 2:00PM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 2:00PM - 4/05/2015 2:15PM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 2:15PM - 4/05/2015 2:30PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 2:30PM - 4/05/2015 3:00PM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 3:00PM - 4/05/2015 3:15PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 3:15PM - 4/05/2015 3:30PM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 3:30PM - 4/05/2015 3:45PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 3:45PM - 4/05/2015 4:15PM"/>
<WorkloadDetail headcount="6" startAndEnd="4/05/2015 4:15PM - 4/05/2015 4:30PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 4:30PM - 4/05/2015 4:45PM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 4:45PM - 4/05/2015 5:15PM"/>
<WorkloadDetail headcount="5" startAndEnd="4/05/2015 5:15PM - 4/05/2015 5:45PM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 5:45PM - 4/05/2015 6:45PM"/>
<WorkloadDetail headcount="3" startAndEnd="4/05/2015 6:45PM - 4/05/2015 8:00PM"/>
<WorkloadDetail headcount="4" startAndEnd="4/05/2015 8:00PM - 4/05/2015 8:15PM"/>
<WorkloadDetail headcount="3" startAndEnd="4/05/2015 8:15PM - 4/05/2015 9:30PM"/>
<WorkloadDetail headcount="2" startAndEnd="4/05/2015 9:30PM - 4/05/2015 10:15PM"/>
<WorkloadDetail headcount="1" startAndEnd="4/05/2015 10:15PM - 4/05/2015 10:30PM"/>
<WorkloadDetail headcount="2" startAndEnd="4/05/2015 10:30PM - 4/05/2015 11:00PM"/>
<WorkloadDetail headcount="0" startAndEnd="4/05/2015 11:00PM - 4/06/2015 4:30AM"/>
<WorkloadDetail headcount="2" startAndEnd="4/06/2015 4:30AM - 4/06/2015 5:00AM"/>
<WorkloadDetail headcount="3" startAndEnd="4/06/2015 5:00AM - 4/06/2015 5:30AM"/>
<WorkloadDetail headcount="2" startAndEnd="4/06/2015 5:30AM - 4/06/2015 6:00AM"/>
<WorkloadDetail headcount="3" startAndEnd="4/06/2015 6:00AM - 4/06/2015 7:15AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/06/2015 7:15AM - 4/06/2015 7:45AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/06/2015 7:45AM - 4/06/2015 8:45AM"/>
<WorkloadDetail headcount="6" startAndEnd="4/06/2015 8:45AM - 4/06/2015 9:00AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/06/2015 9:00AM - 4/06/2015 9:30AM"/>
<WorkloadDetail headcount="5" startAndEnd="4/06/2015 9:30AM - 4/06/2015 10:00AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/06/2015 10:00AM - 4/06/2015 10:45AM"/>
<WorkloadDetail headcount="3" startAndEnd="4/06/2015 10:45AM - 4/06/2015 11:00AM"/>
<WorkloadDetail headcount="4" startAndEnd="4/06/2015 11:00AM - 4/06/2015 11:45AM"/>
```


Figure B.2 Extrait de demande pour l'activité 14768

Durée	Employé	Début	Job
8:00	99232	4/07/2015 4:30AM	14765
4:30	99232	4/08/2015 4:30AM	14765
6:30	99232	4/09/2015 4:30AM	14765
7:30	99232	4/10/2015 4:30AM	14765
4:00	99232	4/11/2015 5:00AM	14765
6:15	238230	4/05/2015 10:15AM	14768
4:00	238230	4/10/2015 8:00PM	14768
4:45	238230	4/11/2015 2:45PM	14768
4:15	104628	4/06/2015 8:45AM	14768
4:15	104628	4/07/2015 9:15AM	14768
4:45	104628	4/09/2015 11:00AM	14768
4:45	104628	4/10/2015 4:30AM	14768
5:45	104628	4/11/2015 8:15AM	14768
6:30	100511	4/05/2015 12:30PM	14765
5:00	100511	4/06/2015 2:00PM	14765
6:30	100511	4/07/2015 12:30PM	14765
8:00	100511	4/09/2015 11:00AM	14765
4:30	100511	4/10/2015 7:30PM	14765
8:00	100511	4/11/2015 9:00AM	14765
7:00	100758	4/05/2015 5:30AM	14765
5:30	100758	4/06/2015 8:30AM	14765
8:00	100758	4/08/2015 9:00AM	14765
4:30	100758	4/09/2015 12:00PM	14768
7:30	100758	4/10/2015 12:00PM	14765
6:00	100758	4/11/2015 5:00PM	14765

Durée	Employé	Début	Job
8:00	99232	4/07/2015 5:00AM	14768
8:00	99232	4/08/2015 5:00AM	14768
8:00	99232	4/09/2015 6:00AM	14768
8:00	99232	4/10/2015 5:00AM	14768
8:00	99232	4/11/2015 8:30AM	14768
8:00	238230	4/05/2015 8:30AM	14768
8:00	238230	4/10/2015 2:30PM	14768
8:00	238230	4/11/2015 10:00AM	14768
6:00	104628	4/06/2015 8:30AM	14768
6:00	104628	4/07/2015 8:30AM	14768
7:30	104628	4/09/2015 8:30AM	14768
8:00	104628	4/10/2015 7:30AM	14768
8:00	104628	4/11/2015 7:00AM	14768
8:00	100511	4/05/2015 10:00AM	14768
7:15	100511	4/06/2015 2:15PM	14768
4:15	100511	4/07/2015 2:30PM	14768
5:45	100511	4/08/2015 11:45AM	14768
7:15	100511	4/09/2015 2:30PM	14768
7:30	100511	4/11/2015 4:30PM	14768
8:00	100758	4/05/2015 6:00AM	14768
4:45	100758	4/06/2015 12:30PM	14768
8:00	100758	4/07/2015 3:00PM	14768
8:00	100758	4/08/2015 7:30AM	14768
7:15	100758	4/09/2015 3:00PM	14768
4:00	100758	4/11/2015 3:30PM	14768

Figure B.3 Différence d'attribution des activités entre la résolution sans (à gauche) et avec préférence (à droite)